

**PREDIKSI PERTUMBUHAN AWAN CUMULONIMBUS PADA CITRA
HIMAWARI IR ENHANCED MENGGUNAKAN *DEEP ECHO STATE*
*NETWORK (DEEPESN)***

SKRIPSI



**UIN SUNAN AMPEL
S U R A B A Y A**

Disusun Oleh
MUHAMMAD FAHRUR ROZI
H02216010

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA**

2019

**PREDIKSI PERTUMBUHAN AWAN CUMULONIMBUS PADA CITRA
HIMAWARI IR ENHANCED MENGGUNAKAN DEEP ECHO STATE
NETWORK (DEEPESN)**

SKRIPSI

Diajukan guna memenuhi salah satu persyaratan untuk memperoleh gelar Sarjana Matematika (S.Mat) pada Program Studi Matematika



**UIN SUNAN AMPEL
S U R A B A Y A**

Disusun oleh
MUHAMMAD FAHRUR ROZI
H02216010

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA**

2019

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini,

Nama : Muhammad Fahrur Rozi

NIM : H02216010

Program Studi : Matematika

Angkatan : 2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul "Prediksi Pertumbuhan Awan Cumulonimbus pada Citra Himawari IR Enhanced Menggunakan Deep Echo State Network (DeepESN)". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 23 Desember 2019

Yang menyatakan,



Muhammad Fahrur Rozi

NIM. H02216010

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh

Nama : Muhammad Fahrur Rozi

NIM : H02216010

Judul Skripsi : Prediksi Pertumbuhan Awan Cumulonimbus pada Citra
Himawari IR Enhanced Menggunakan Deep Echo State
Network (DeepESN)

telah diperiksa dan disetujui untuk diujikan.

Surabaya, 23 Desember 2019

Pembimbing



Dian Candra Rini Novitasari, M.Kom
NIP. 198511242014032001

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi oleh

Nama : Muhammad Fahrur Rozi
NIM : H02216010
Judul Skripsi : Prediksi Pertumbuhan Awan Cumulonimbus pada Citra Himawari IR Enhanced Menggunakan *Deep Echo State Network* (DeepESN)

Telah dipertahankan di depan Tim Penguji
pada tanggal Desember 2019

Mengesahkan,
Tim Penguji

Penguji I

Dian Candra Rini Novitasari, M.Kom
NIP. 198511242014032001

Penguji II

Nurissaideh Ulinnuha, M.Kom
NIP. 199011022014032004

Penguji III

Aris Fanani, M.Kom
NIP. 198701272014031002

Penguji IV

Wika Dianita Utami, M.Sc
NIP. 199206101018012003

Mengetahui,

Dekan Fakultas Sains dan Teknologi
DIN Suhar Ampel Surabaya





**KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN**

Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : MUHAMMAD FAHRUR ROZI
NIM : H02216010
Fakultas/Jurusan : SAINS / MATEMATIKA
E-mail address : m.f.rozi.mfr@gmail.com

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :

Sekripsi Tesis Desertasi Lain-lain (.....)
yang berjudul :

PREDIKSI PERTUMBUHAN AWAN COMULONIMBUS PADA CITRA HIMAWARI IR ENHANCED

MENGGUNAKAN DEEP ECHO STATE NETWORK (DeepESN)

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara **fulltext** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 30 DESEMBER 2019

Penulis

(MUHAMMAD FAHRUP ROZI)
nama terang dan tanda tangan .

ABSTRAK

Prediksi Pertumbuhan Awan Cumulonimbus pada Citra Himawari IR

Enhanced Menggunakan Deep Echo State Network (DeepESN)

Awan Cumulonimbus (Cb) merupakan awan yang memiliki suhu sangat rendah. Suhu yang dimiliki oleh awan Cb dapat mencapai -100° C. Suhu dalam awan Cb memicu turunnya sebuah hujan es. Aktivitas *updraft*, dan *downdraft* memicu munculnya badai petir yang ada dalam sebuah awan Cb. Dibutuhkan sebuah pengamatan awan untuk memberi peringatan ketika tumbuh awan dengan suhu yang cukup rendah. Citra Himawari IR *Enhanced* merupakan citra satelit yang menampilkan suhu awan yang sedang diamati. Sebagai upaya untuk mengetahui kondisi awan selanjutnya diperlukan sebuah sistem prediksi. Salah satu sistem prediksi yang diusulkan yaitu *Deep Echo State Network* (DeepESN). Metode DeepESN merupakan salah satu metode yang dapat mempelajari pola data berdasarkan waktu. DeepESN yang termasuk dalam Deep Learning, memiliki kemampuan untuk mempelajari sebuah citra tanpa melalui proses ekstraksi fitur. Sebagai tolak ukur sistem yang dibuat DeepESN baik digunakan evaluasi kemiripan citra *Peak Signal-to-Noise Ratio* (PSNR). Setelah dilakukan beberapa percobaan didapatkan model DeepESN terbaik dengan *learning rate* 0.5, jumlah node 120, dan 150 lapisan *reservoir*. Pada percobaan tersebut menghasilkan hasil prediksi dengan rata-rata tingkat kemiripan citra sebesar 22.994 dB. Hasil terbaik dengan DeepESN untuk memprediksi satu citra dapat mencapai 51.2006 dB.

Kata kunci: Cumulonimbus (Cb), Citra Himawari IR Enhanced, Deep Echo State Network (DeepESN), Peak Signal-to-Noise Ratio (PSNR)

ABSTRACT

Growth Prediction for Cumulonimbus Clouds in Himawari IR Enhanced Image Using the Deep Echo State Network (DeepESN)

Cumulonimbus clouds (Cb) are clouds that have very low temperatures. The temperature of the Cb cloud can reach -100°C . The temperature in the Cb cloud triggers hail. The activity of *updraft* and *downdraft* triggered the appearance of a thunderstorm in a Cb cloud. A cloud observation is needed to warn when a cloud is growing at a low enough temperature. The Himawari IR *Enhanced* image is a satellite image that displays the temperature of the cloud being observed. In an effort to find out the next cloud condition, a prediction system is needed. One of the proposed prediction systems is *Deep Echo State Network* (DeepESN). DeepESN method is one method that can study data patterns based on time. DeepESN which is included in Deep Learning, has the ability to learn an image without going through the feature extraction process. As a system benchmark made by DeepESN, evaluate the similarity of the image using the *Peak Signal-to-Noise Ratio* (PSNR) method. After several experiments, the best DeepESN model with *learning rate* 0.5, number of node 120, and 150 layers *reservoir* was obtained. In these experiments produce prediction results with an average image similarity level of 22.994 dB. The results of predictions with DeepESN to obtain an image that has a PSNR value of 51.2006 dB

Keywords: Cumulonimbus (Cb), Himawari IR Enhanced Image, Deep Echo State Network (DeepESN), Peak Signal-to-Noise Ratio (PSNR)

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN PEMBIMBING	ii
PENGESAHAN TIM PENGUJI SKRIPSI	iii
HALAMAN PERNYATAAN KEASLIAN	iv
MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR TABEL	xi
ABSTRAK	xii
ABSTRACT	xiii
I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	6
1.3. Tujuan Penelitian	6
1.4. Manfaat Penelitian	6
1.5. Batasan Masalah	7
1.6. Sistematika Penulisan	7
II TINJAUAN PUSTAKA	9
2.1. Citra Digital	9
2.2. Satelit	11
2.3. Citra Satelit	13
2.3.1. Himawari-8 IR Enhanced	14
2.3.2. Himawari-8 Natural Color	14
2.3.3. Citra Himawari-8 WV Enhanced	15
2.3.4. Himawari-8 Rainfall Potential (RP)	16

2.3.5. Citra <i>Hotspot</i> MODIS	17
2.3.6. Citra Himawari-8 <i>GeoHotspot</i>	18
2.4. Awan	19
2.5. <i>Recurrent Neural Networks</i> (RNNs)	22
2.6. <i>Reservoir Computing</i> (RC)	25
2.7. <i>Echo State Network</i> (ESN)	27
2.8. <i>Deep Echo State Networks</i> (DeepESN)	32
2.9. <i>Peak Signal-to-Noise Ratio</i> (PSNR)	34
III METODE PENELITIAN	36
3.1. Data Penelitian	36
3.2. Rancangan Penelitian	36
IV HASIL DAN PEMBAHASAN	41
4.1. <i>Preprocessing</i> Data	41
4.2. Prediksi Menggunakan <i>Deep Echo State Network</i> (DeepESN)	47
V PENUTUP	66
5.1. Simpulan	66
5.2. Saran	67
DAFTAR PUSTAKA	68

DAFTAR TABEL

4.1	Contoh Piksel Citra Ukuran 10×10	44
4.2	Piksel Ketetangan dari Nilai Tabel 4.1 Baris 2, dan Kolom 2	45
4.3	Nilai Piksel Setelah diurutkan	45
4.4	Nilai Piksel Citra Setelah Diberikan Median Filter	45
4.5	Data Pada Matriks U	46
4.6	Nilai Matriks Bobot Masukan (w_{in})	48
4.7	Nilai Matriks $w_{l=1}$	50
4.8	Nilai Matriks $w_{l=2}$	50
4.9	Nilai Matriks $w_{l=99}$	51
4.10	Nilai Matriks w_x Pada Lapisan 1	51
4.11	Nilai Matriks w_x Pada Lapisan 2	52
4.12	Nilai Matriks w_x Pada Lapisan 100	52
4.13	Nilai Matriks w_θ	56
4.14	Hasil Rata Rata PSNR dari Beberapa Percobaan	61
4.15	Sebaran Nilai PSNR pada Percobaan ke 5	63

DAFTAR GAMBAR

2.1	Citra Himawari-8 IR <i>Enhanced</i>	14
2.2	Citra Himawari-8 <i>Natural Color</i>	15
2.3	Citra Himawari-8 WV <i>Enhanced</i>	16
2.4	Citra Himawari-8 <i>Rainfall Potential (RP)</i>	17
2.5	Citra Hotspot MODIS	18
2.6	Citra Himawari-8 <i>GeoHotspot</i>	18
2.7	Struktur <i>Recurrent Neural Networks (RNNs)</i>	22
2.8	Struktur Jaringan <i>Echo State Network (ESN)</i>	28
2.9	Struktur <i>Echo State Network (ESN)</i> Menggunakan Bobot fb	30
2.10	Struktur <i>Deep Echo State Network (DeepESN)</i>	32
3.1	Diagram Alir Rancangan Penelitian	37
3.2	Diagram Alir Pelatihan DeepESN	39
4.1	Contoh Data Citra Himawari-8 IR <i>Enhanced</i>	42
4.2	Contoh Hasil Pemotongan Citra pada Wilayah Kalimantan Selatan, dan Sekitarnya	43
4.3	Contoh Hasil Pengambilan Citra Lapisan Merah	44
4.4	Contoh Hasil Penghilangan <i>Noise</i> Menggunakan Median Filter	45
4.5	Sebaran nilai PSNR Pada Percobaan ke 5	62
4.6	Citra dengan Nilai PSNR Rendah	64
4.7	Citra Prediksi yang Mempunyai Kemiripan dengan Citra Aslinya	64

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Negara Indonesia merupakan negara kepulauan. Jumlah pulau yang dimiliki Indonesia mencapai 17499 pulau. Pulau-pulau yang ada di Indonesia dipisahkan oleh lautan yang luas. Luas laut yang dimiliki oleh Indonesia lebih luas jika dibandingkan dengan luas daratan. Perbandingan laut dan daratan Indonesia yaitu 3:2. Luas lautan Indonesia membuat Indonesia dapat disebut dengan negara maritim. Luas laut Indonesia 3.25 juta km^2 , sedangkan luas daratan Indonesia 2.01 juta km^2 , sisanya 2.55 juta km^2 sebagai Zona Ekonomi Eksklusif (ZEE). Jika dilihat dari letak geografisnya, Indonesia terletak di daerah khatulistiwa. Letak Indonesia di daerah khatulistiwa menyebabkan Indonesia mendapatkan penyinaran matahari sepanjang tahun (Roza, 2019).

Penyinaran matahari sepanjang tahun dan wilayah laut yang sangat luas menyebabkan tingginya proses penguapan air. Penguapan air yang tinggi berimbang pada proses pertumbuhan awan yang tinggi. Selain itu posisi Indonesia dalam garis khatulistiwa menjadi garis belok angin. Garis belok angin menjadikan titik kumpul uap air terkumpul di negara Indonesia. Kumpulan uap air terkumpul menjadi berbagai macam awan. Awan yang terbentuk dapat berlangsung singkat atau terbentuk dengan waktu yang cukup lama. Pembentukan awan yang lama memicu tumbuhnya awan yang berpotensi hujan. Selain itu tumbuhnya awan yang sangat besar dan padat memicu munculnya awan cumulonimbus (Handoyo, 2011).

Awan cumulonimbus (Cb), merupakan awan yang memiliki pertumbuhan secara vertikal. Awan Cb memiliki kandungan uap air yang sangat tinggi. Aktivitas dalam awan Cb sering terjadi badai petir. Badai petir dalam sebuah awan Cb terbentuk karena proses *updraft*, dan *downdraft*. Proses *updraft* dan *downdraft* menimbulkan gesekan antar partikel yang naik dengan partikel yang turun. Gesekan kedua partikel menimbulkan petir yang terjadi dalam sebuah awan Cb. Kandungan dalam awan Cb memiliki suhu yang sangat rendah. Sebuah awan Cb dapat memiliki suhu -100°C . Suhu dingin yang dimiliki oleh awan Cb menyebabkan uap air yang terkumpul berubah menjadi butiran es (Emiliani, 1992).

Kandungan dalam sebuah awan Cb menyebabkan gangguan pada beberapa aktivitas manusia. Badai petir yang terjadi dalam awan Cb mengharuskan pengalihan jalur penerbangan. Kandungan es pada awan Cb membahayakan pesawat karena butiran es dapat masuk dalam mesin pesawat. Selain itu suhu dingin awan Cb dapat membekukan mesin pesawat, sehingga pesawat tidak dapat berfungsi secara normal. Munculnya awan Cb harus diketahui seorang pilot sebelum melakukan penerbangan. Seorang pilot harus menghindari awan Cb dari jauh karena ukuran yang dimiliki awan Cb sangat tinggi dan lebar (PaskiPermana, Sepriando, & Pertiwi, 2017).

Bahaya awan Cb dapat mengganggu aktivitas manusia yang ada di daratan. Suhu udara yang tidak mampu mencairkan es menimbulkan terjadinya hujan es. Selain itu hujan yang dihasilkan awan Cb sangat lebat, hingga seringkali disertai badai dan angin kencang. Gangguan yang disebabkan oleh awan Cb, menjadikan manusia harus memperhitungkan pertumbuhan dari awan Cb. Pertumbuhan awan Cb digunakan untuk memperkirakan dan merencanakan aktivitas yang harus dilakukan sebelum tumbuhnya awan Cb. Perkiraan tumbuhnya awan Cb dapat

dimanfaatkan untuk proses industri, perdagangan, maupun untuk menjaga kesehatan manusia (PaskiPermana, Sepriando, & Pertiwi, 2017). Proses pembentukan dan kandungan awan dijelaskan pula dalam salah satu ayat *Al Quran*. Surat *An-Nur* ayat 43 menjelaskan bahwa sebuah awan telah digerakkan untuk saling menumpuk satu sama lain. Sebuah awan terbentuk dari air dan butiran butiran es. Butiran es turun dari selah selah awan menjadi sebuah hujan. Kandungan es dalam awan menjadi salah satu bahaya yang dapat mengganggu manusia.

Sebagai bentuk antisipasi yaitu memperkirakan proses pertumbuhan awan Cb. Perkiraan pertumbuhan awan Cb dapat melihat dari penangkapan citra satelit. Salah satu citra satelit yang dapat menggambarkan keberadaan awan Cb yaitu citra Himawari-8 IR *Enhanced*. Citra IR *Enhanced* memberikan gambaran awan Cb dengan menunjukkan suhu titik puncak awan. Sebuah awan yang terbaca dengan suhu kurang dari -60°C merupakan awan Cb. Suhu puncak awan yang melebihi suhu tersebut merupakan suhu yang dimiliki oleh jenis awan lainnya. Tidak menutup kemungkinan bahwa tumbuhnya awan Cb dikarenakan pertumbuhan awan yang lain. Proses pertumbuhan awan Cb dapat dilihat dengan menurunya suhu awan hingga kurang dari -60°C . Penurunan suhu sebuah awan dapat diprediksi untuk memperkirakan suhu awan di waktu mendatang (Efendi & Fachrurrozi, 2017).

Proses prediksi dilakukan dengan mengenali pola pertumbuhan awan dari sebuah citra. Pada penelitian ini pengenalan pola didapatkan dari sebuah citra satelit Himawari-8 IR *Enhanced*. Pengenalan pola dari sebuah citra satelit diambil dengan melihat nilai piksel yang ada pada citra satelit. Nilai piksel menunjukkan suhu yang dimiliki oleh awan. Semakin tinggi nilai piksel pada intensitas merah maka

semakin dingin suhu awan. Awan dengan suhu yang lebih tinggi memiliki intensitas warna merah yang rendah. Suhu awan pada satu titik tertentu sangat berpengaruh pada suhu awan di sekelilingnya. Akibatnya, sebuah piksel akan berpengaruh pada piksel yang ada di sekitarnya. Penentuan nilai piksel pada waktu t diperoleh dari piksel sekitarnya di waktu $t - 1$. Sebuah sistem untuk mempelajari kondisi tersebut sangatlah dibutuhkan (Pandjaitan & Damayanti, 2018).

Salah satu metode yang mempelajari pola dari waktu ke waktu yaitu *Echo State Network* (ESN). Metode ESN mempelajari pola di waktu sebelumnya untuk mempertimbangkan nilai keluaran pada waktu mendatang. Proses yang dimiliki oleh ESN seperti proses yang berjalan pada *Recurrent Neural Networks* (RNNs). Perbedaan dari RNN yaitu proses pelatihan yang dimiliki oleh ESN. Pelatihan RNN dilakukan untuk setiap lapisan, sedangkan pelatihan ESN hanya berlaku pada lapisan *readout*. Pelatihan ESN berdasarkan pada pengembangan dari proses *Reservoir Computing* (RC). Peningkatan kinerja ESN dikembangkan dengan melakukan beberapa modifikasi pada jaringan ESN (Ma et al., 2016).

Sebagai pengembangan dari pembelajaran ESN, dibentuk lapisan berlapis menggunakan jaringan ESN. Proses penumpukan ESN secara berlapis disebut dengan *Deep Echo State Network* (DeepESN). Prediksi menggunakan DeepESN memiliki vektor masukan dalam sebuah lapisan *reservoir*. Nilai keluaran dari setiap lapisan *reservoir* dipetakan menjadi satu pada lapisan *readout*. Sama seperti proses pada ESN, pelatihan bobot dilakukan hanya pada lapisan menuju *readout*. Nilai bobot yang dilatih pada lapisan *readout* menjadi nilai pembelajaran pola. Pola yang dihasilkan *readout* menjadi pola untuk melakukan prediksi pada waktu mendatang. Pada setiap waktu nilai vektor masukan dipelajari untuk mempelajari pola di waktu mendatang (C. S. Lee, Baughman, & Lee, 2017).

Hasil penelitian sebelumnya menyatakan kinerja DeepESN memiliki kinerja yang lebih optimal jika dibandingkan metode *Reservoir Computing* (RC) sederhana (Long, Zhang, & Li, 2019). Penelitian sebelumnya menyatakan bahwa penyimpanan memori pada DeepESN lebih tinggi daripada ESN. Penyimpanan DeepESN lebih unggul 20% lebih dari metode ESN sederhana (Gallicchio & Micheli, 2018). DeepESN memiliki kinerja yang baik untuk mengidentifikasi kesalahan secara maksimal dan akurat. Proses prediksi menggunakan DeepESN bertujuan untuk mendapatkan nilai fitur awan di waktu mendatang. Jumlah fitur yang dikeluarkan oleh DeepESN sama seperti ukuran fitur dari satu citra masukan (Liu, Chen, Yin, & Saad, 2018).

Sebagai evaluasi dari hasil prediksi dibutuhkan sistem untuk mengenali kemiripan citra. *Peak signal-to-noise ratio* (PSNR) merupakan salah satu metode yang dapat mempelajari kemiripan antara 2 citra. PSNR dapat mengenali kemiripan menggunakan perhitungan eror pada setiap piksel citra. Penelitian sebelumnya menggunakan PSNR untuk melihat kualitas segmentasi citra plat kendaraan (PirahanSiah, Abdullah, & Sahran, 2010). Penerapan metode PSNR juga diterapkan untuk mengevaluasi kemiripan citra sebelum dan sesudah dilakukan pengolahan citra (Yao, Lin, Ong, & Lu, 2005). Penelitian ini menggunakan PSNR sebagai nilai evaluasi dari citra data sebenarnya dengan citra hasil prediksi.

Melihat dari beberapa penjelasan sebelumnya menjadikan penelitian dengan judul "Predisi Pertumbuhan Awan Cumulonimbus pada Data Satelit Himawari IR Enhanced Menggunakan Metode *Deep Echo State Network* (DeepESN)". Proses prediksi pada data citra IR *Enhanced*. Sebagai bentuk evaluasi sistem digunakan metode PSNR. Evaluasi ditujukan untuk melihat kondisi waktu mendatang akan

tumbuh awan Cb sesuai dengan data yang diujikan. Hasil yang diperoleh pada akhirnya dapat membantu untuk penanganan masalah kedatangan awan Cb.

1.2. Rumusan Masalah

Berdasarkan penjelasan pada bagian latar belakang, peneliti memiliki beberapa rumusan masalah sebagai berikut.

1. Bagaimana cara melakukan prediksi pertumbuhan awan cumulonimbus menggunakan *Deep Echo State Network*?
 2. Bagaimana tingkat keberhasilan suatu sistem *Deep Echo State Network* untuk melakukan prediksi terhadap pertumbuhan awan cumulonimbus?

1.3. Tujuan Penelitian

1. Mengetahui cara melakukan prediksi pertumbuhan awan cumulonimbus menggunakan *Deep Echo State Network*.
 2. Mengetahui tingkat keberhasilan suatu sistem *Deep Echo State Network* untuk melakukan prediksi terhadap pertumbuhan awan cumulonimbus.

1.4. Manfaat Penelitian

- #### 1. Bagi Mahasiswa dan Peneliti lainnya

Menambah pengetahuan tentang cara melakukan prediksi pertumbuhan awan cumulonimbus menggunakan *Deep Echo State Network*.

- ## 2. Bagi Masyarakat

Meningkatkan kewaspadaan masyarakat ketika terdapat pertumbuhan awan cumulonimbus yang sewaktu waktu dapat muncul.

1.5. Batasan Masalah

1. Penelitian ini dilakukan menggunakan data berupa citra Himawari IR Enhanced.
 2. Data yang digunakan merupakan data yang diambil setiap satu jam sekali.
 3. Penelitian ini melakukan terbentuknya awan di daerah Kalimantan Selatan dan sekitarnya.

1.6. Sistematika Penulisan

Sistematika penyusunan yang digunakan pada penulisan laporan kerja lapangan ini sebagai berikut.

1. BAB I : PENDAHULUAN

Bab ini menjelaskan latar belakang penulis untuk melakukan penelitian. Selain itu bab ini juga menjelaskan bagaimana rumusan masalah dan tujuan penelitian. Penyelesaian yang diharapkan oleh seorang peneliti dapat memberi manfaat yang akan dijelaskan pada bagian manfaat penelitian.

2. BAB II : DASAR TEORI

Bab ini berisikan tentang teori-teori yang mendukung penyelesaian penelitian. Teori yang digunakan merupakan teori yang diambil dari penelitian sebelumnya. Kumpulan teori oleh peneliti sebelumnya dapat dijadikan sebagai dukungan untuk melakukan prediksi pertumbuhan awan cumulonimbus pada citra Himawari IR Enhanced menggunakan *Deep Echo State Network*.

3. BAB III : METODE PENELITIAN

Bab ini menjelaskan gambaran umum peneliti yang digunakan untuk menyelesaikan prediksi pertumbuhan awan cumulonimbus pada citra Himawari

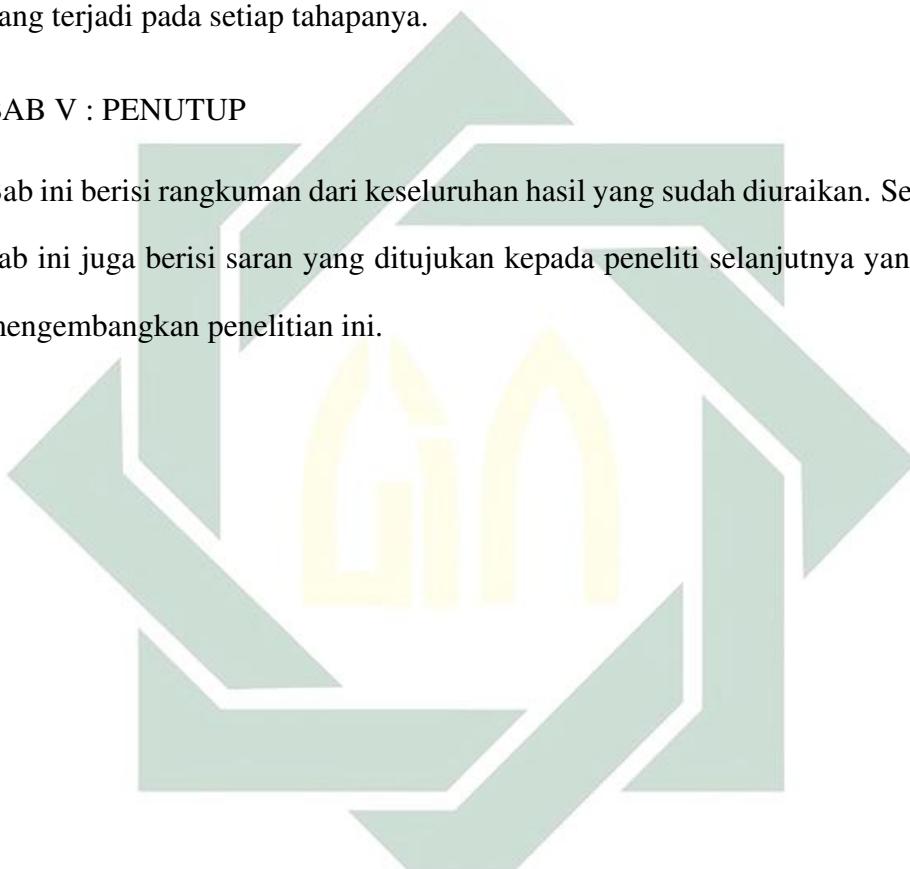
IR Enhanced menggunakan *Deep Echo State Network*

4. BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisikan hasil penelitian yang sudah dilakukan peneliti. Selain itu pembahasan dari hasil hasil penelitian dijabarkan untuk menjelaskan proses yang terjadi pada setiap tahapanya.

5. BAB V : PENUTUP

Bab ini berisi rangkuman dari keseluruhan hasil yang sudah diuraikan. Selain itu bab ini juga berisi saran yang ditujukan kepada peneliti selanjutnya yang ingin mengembangkan penelitian ini.



BAB II

TINJAUAN PUSTAKA

2.1. Citra Digital

Citra merupakan sebuah matriks yang memiliki nilai real. Setiap nilai real yang ada dalam matriks merupakan representasi dari intensitas cahaya yang ada. Semakin tinggi intensitas cahaya yang ditangkap semakin tinggi nilai yang dimilikinya. Citra merupakan matriks dengan ukuran $M \times N$, dengan M merupakan banyak baris, dan N merupakan banyak kolom dari matriks. Matriks sebuah citra didefinisikan sebagai sebuah fungsi $f(x, y)$ dengan x adalah koordinat pada baris, dan y merupakan koordinat posisi pada kolom matriks. Setiap nilai dalam matriks citra disebut dengan suatu piksel (Kadir & Susanto, 2013). Bentuk dari fungsi citra disajikan pada Persamaan 2.1.

$$f(x, y) = \begin{bmatrix} f(1, 1) & f(1, 2) & \cdots & f(1, N) \\ f(2, 1) & f(2, 2) & \cdots & f(2, N) \\ \vdots & \vdots & & \vdots \\ f(M, 1) & f(M, 2) & \cdots & f(M, N) \end{bmatrix} \quad (2.1)$$

Jika dilihat dari nilai piksel pada citra, umumnya citra digolongkan menjadi tiga jenis. Diantara jenis citra tersebut yaitu citra biner, citra *grayscale*, dan citra RGB. Citra biner merupakan citra dengan nilai piksel antara nol atau satu. Nilai nol merepresentasikan warna hitam, sedangkan nilai satu menunjukkan warna putih. Jenis citra yang kedua yaitu citra yang melihat derajat keabuan atau biasa disebut

dengan cira *grayscale*. Berbeda dengan citra biner, citra jenis ini memiliki interval nilai antara 0 sampai 255. Semakin tinggi nilai piksel yang dimiliki maka semakin terang cahaya yang ditangkap. Jenis terakhir yaitu citra RGB. Citra ini memiliki tiga matriks masing-masing matriks mewakili intensitas cahaya berwarna merah, hijau, dan biru. Citra jenis ini menghasilkan citra berwarna, bergantung pada kombinasi dari ketiga warna tersebut (Kadir & Susanto, 2013).

Melalui kemajuan teknologi sebuah citra dapat digunakan untuk mendapatkan berbagai macam informasi. Salah satu cara untuk mendapatkan informasi dari citra yaitu dengan melakukan pengolahan pada citra tersebut. Pengolahan citra digital merupakan pengolahan yang dilakukan menggunakan komputer atau sistem digital. Dilihat dari keluarannya, pengolahan citra digital dibagi menjadi tiga tingkatan yaitu tingkatan rendah (*low*), menengah (*middle*), tinggi (*high*). Pengolahan citra tingkat rendah merupakan pengolahan citra seperti pada tahap pra-pemrosesan citra. Beberapa pengolahan tingkat rendah yang sering dilakukan yaitu proses penghilangan *noise*, peningkatan atau penurunan kontras, dan penajaman gambar. Pengolahan tahap ini dilakukan dengan masukan, dan keluaran berupa citra(Putra, 2010). Pengolahan citra tingkat menengah dilakukan pada proses seperti segmentasi, deskripsi objek, dan klasifikasi objek pada citra. Pada pengolahan tingkat menengah proses yang dilakukan menggunakan masukan berupa citra, dengan keluaran berupa nilai ekstraksi. Beberapa contoh keluaran yang dapat dihasilkan yaitu berupa tepi, kontur, atau indentitas pada suatu citra. Sedangkan pengolahan tingkat tinggi merupakan pengolahan berupa analisa dari hasil pengolahan citra menengah (Putra, 2010).

2.2. Satelit

Awalnya satelit merupakan nama yang digunakan untuk sebuah benda yang mengelilingi bumi. Setiap planet memiliki jumlah satelit berbeda-beda. Jumlah satelit yang mengelilingi bumi hanya satu yaitu bulan. Sedangkan jumlah satelit pada venus yaitu 4, dan planet merkurius 8 satelit. Planet bumi juga menjadi satelit terhadap benda langit lainnya yang lebih besar. Sebagai contoh bumi menjadi satelit matahari, dan mengelilingi matahari selama 365,25 hari. Menggunakan teknologi, dan pemikiran yang semakin maju, membuat sebuah ide meluncurkan satelit buatan untuk mengelilingi bumi. Pada tahun 1945 merupakan tahun pertama peluncuran satelit buatan pada orbit ekuator. Peluncuran satelit pertama kalinya dicetuskan oleh Arthur C Clarke (Dewi & Supianto, 2015).

Melihat dari keberhasilan satelit yang diluncurkan, membuat peluncuran satelit semakin hari semakin bertambah. Jumlah satelit yang diluncurkan sangat banyak. Setiap satelit yang diluncurkan memiliki ketinggian, garis edar, dan fungsi yang berbeda-beda. Jika dilihat dari orbitnya satelit dibedakan menjadi 3 jenis. Jenis yang pertama yaitu satelit stasioner. Satelit stasioner diluncurkan, dan diletakkan pada titik yang tetap. Satelit stasioner memperoleh titik yang tetap dengan mengitari bumi sesuai dengan arah rotasi bumi. Satelit stasioner memiliki arah orbit dari timur menuju barat. Peluncuran satelit stasioner memiliki ketinggian yang bermacam macam. Berikut jenis satelit stasioner jika dilihat dari ketinggiannya (Imam & Pamungkas, 2014).

1. *Low Earth Orbit (LEO)*

Satelit LEO memiliki ketinggian 320 sampai 800 kilometer di atas permukaan bumi. Letaknya yang masih dekat dengan bumi, membuat satelit LEO harus memiliki kecepatan yang sangat tinggi. Sebuah satelit LEO dapat memiliki

kecepatan edar mencapai 27359 km/jam . Kecepatan yang sangat tinggi membuat satelit LEO dapat mengelilingi bumi hanya dalam waktu 90 menit. Jumlah satelit LEO yang beredar mengelilingi bumi lebih dari 8000 satelit (Imam & Pamungkas, 2014).

2. *Medium Earth Orbit (MEO)*

Satelit MEO berada sedikit lebih tinggi dibandingkan dengan satelit LEO. Satelit MEO memiliki ketinggian 10000 kilometer di atas permukaan bumi. Jumlah satelit MEO lebih sedikit jika dibandingkan dengan satelit LEO (Imam & Pamungkas, 2014).

3. *Geostationary Earth Orbit (GEO)*

Satelit GEO memiliki ketinggian 35786 kilometer di atas permukaan bumi. Garis edar yang dimiliki oleh satelit GEO berada pada garis khatulistiwa. Sedangkan kecepatan yang dimiliki sebuah satelit GEO untuk mengelilingi bumi sama seperti waktu bumi mengalami rotasi (Imam & Pamungkas, 2014).

Setiap satelit yang diluncurkan memiliki tujuan yang berbeda-beda. Semua satelit yang diluncurkan bertujuan untuk memberikan pelayanan terhadap kebutuhan manusia. Jika dilihat dari pelayanan yang diberikan terdapat beberapa empat jenis satelit. Jenis pertama yaitu *fixed service satellite*. Satelit jenis ini digunakan untuk memberikan layanan telpon, transit data (internet), dan TV *broadcasting*. Satelit jenis kedua yaitu *direct broadcast satellite*. Satelit ini bertujuan khusus untuk TV *broadcasting*, dan radio. *Mobile satellite service* merupakan jenis satelit yang ketiga. Satelit jenis ketiga ditujukan khusus untuk pelayanan telpon nirkabel. Satelit jenis terakhir yaitu *Medium Power Satellite*. Satelit Medium Power digunakan untuk pelayanan umum, dan keperluan militer.

(Dewi & Supianto, 2015).

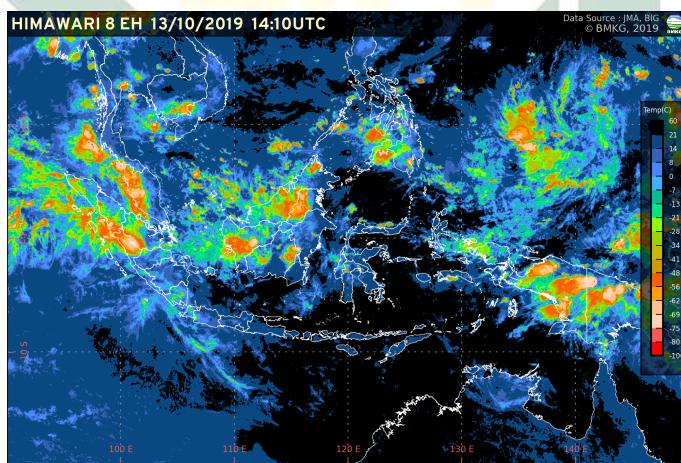
2.3. Citra Satelit

Proses pengolahan citra digital seringkali digunakan untuk membantu kinerja manusia. Hasil pengolahan citra digital memberikan informasi yang dibutuhkan manusia. Pentingnya sebuah pengolahan citra digital membuat pengolahan citra dilakukan pada beberapa bidang. Salah satu bidang yang digunakan dengan memanfaatkan citra digital yaitu pengindraan jarak jauh. Menggunakan pengolahan citra yang maksimal dapat membantu kerja manusia untuk melakukan pengamatan sebuah titik dengan jarak yang cukup jauh. Salah satu bentuk pengolahan citra digital untuk pengindraan jarak jauh adalah citra satelit. Citra satelit dikembangkan untuk melakukan pengamatan visual dengan jarak yang tidak dapat dijangkau oleh pengelihatan manusia. Pengamatan yang dilakukan oleh satelit bukan hanya daerah yang ada di permukaan bumi, melainkan keadaan yang ada di alam semesta. Tingkat pengelihatan yang dimiliki oleh satelit berbeda-beda. Jika dilihat dari resolusinya satelit dibedakan menjadi tiga jenis. Satelit dengan resolusi rendah, menengah, dan tinggi (Suwargana, 2013).

Selain dilihat dari kualitasnya pengamatan citra satelit juga memiliki proses penangkapan citra yang berbeda. Pengambilan citra satelit dapat menggunakan infra merah, pengelihatan secara nyata (visibel), atau yang lainnya. Salah satu satelit yang menghasilkan berbagai macam citra keluaran yaitu satelit himawari. Satelit himawari merupakan satelit yang mengawasi wilayah Indonesia. Hasil pengolahan citra satelit yang dihasilkan oleh satelit himawari bermacam-macam (BMKG, 2019). Beberapa jenis citra yang dihasilkan oleh satelit himawari adalah sebagai berikut:

2.3.1. Himawari-8 IR Enhanced

Citra Himawari-8 IR *Enhanced* merupakan salah satu citra dari satelit himawari yang menunjukkan suhu puncak awan. Penangkapan suhu puncak awan diperoleh dengan memancarkan gelombang 10.4 mikro meter. Hasil pemancaran gelombang kemudian diubah dalam skala suhu antara -100°C hingga 100°C . Selanjutnya nilai suhu puncak awan dikonversi menuju nilai piksel pada sebuah citra dengan interval 0 sampai 255. Citra yang disajikan merupakan citra RGB. Citra dengan warna biru atau hitam menunjukkan tidak terdapat pembentukan awan yang banyak. Jika suatu titik memiliki suhu puncak awan yang semakin tinggi maka warna yang dihasilkan mendekati warna jingga atau merah. Citra dengan warna jingga atau merah menunjukkan pertumbuhan awan yang signifikan, dan sangat memungkinkan terbentuknya awan cumulonimbus (BMKG, 2019). Sebagai contoh citra Himawari-8 IR *Enhanced* ditunjukkan pada Gambar 2.1.

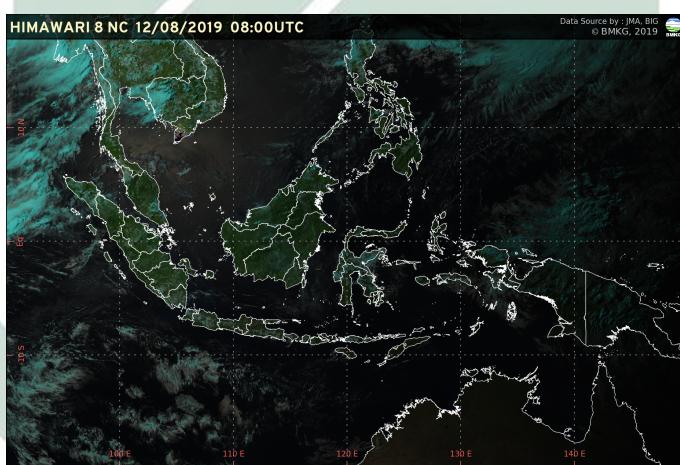


Gambar 2.1 Citra Himawari-8 IR Enhanced

2.3.2. Himawari-8 Natural Color

Citra Himawari-8 *Natural Color* (NC) merupakan citra satelit himawari yang bergantung pada pencahayaan matahari. Citra yang dihasilkan merupakan

citra RGB. Citra jenis ini digunakan untuk mengamati ketebalan awan, *mikrofisis* awan, dan proses konvektifitas. Hasil citra jenis ini merupakan pengamatan yang diperoleh secara nyata atau visibel. Jika tidak ada pencahayaan dari matahari maka satelit tidak dapat menghasilkan citra visibel. Bergantungnya dengan pencahayaan matahari menjadikan pemanfaatan citra ini terbatas. Pengamatan menggunakan citra jenis ini hanya dapat berlangsung sampai sore hari (BMKG, 2019). Produk citra Himawari-8 *Natural Color* (NC) disajikan pada Gambar 2.2.

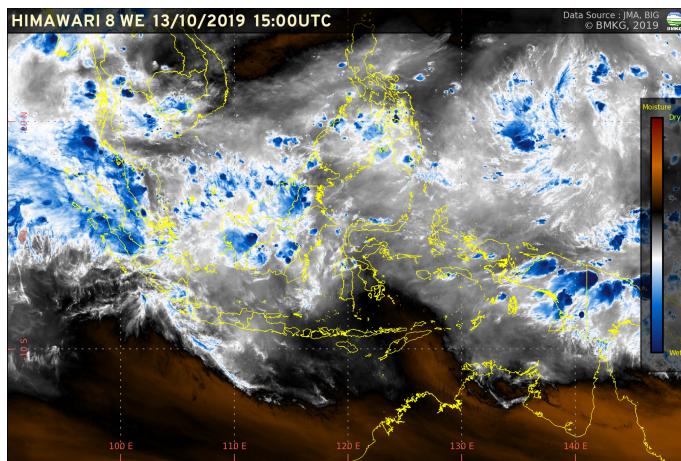


Gambar 2.2 Citra Himawari-8 Natural Color

2.3.3. Citra Himawari-8 WV Enhanced

Citra Himawari-8 WV *Enhanced* merupakan citra satelit himawari yang menggambarkan kelembaban atmosfer. Kelembaban atmosfer yang disajikan yaitu atmosfer tingkat menengah hingga atmosfer atas. Pengambilan citra WV diperoleh menggunakan gelombang 6.2 mikro meter. Hasil citra jenis ini dapat digunakan untuk mengamati kelembaban udara sebagai salah satu faktor dalam proses terbentuknya awan. Warna yang dihasilkan oleh citra WV yaitu antara warna coklat, dan warna biru. Warna coklat menunjukkan keadaan udara yang semakin kering, sedangkan warna biru menunjukkan keadaan udara semakin lembab. Kekurangan citra jenis ini adalah kondisi ketika udara sangat lembab, dan sangat

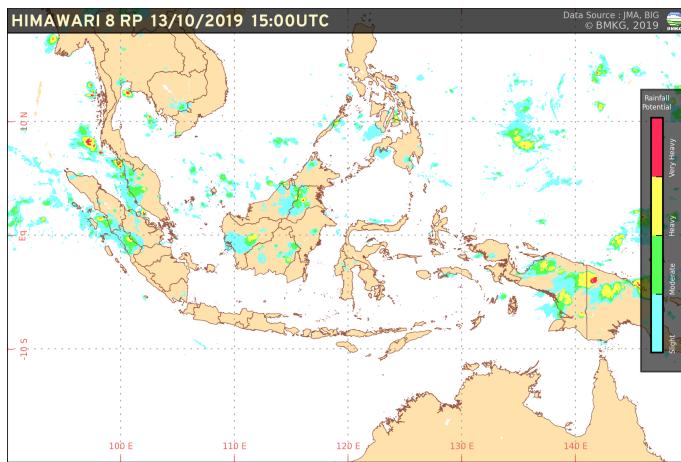
kering memiliki warna yang menuju warna kehitaman. Keadaan tersebut mengakibatkan kedua kondisi yang sangat berbeda namun memiliki nilai yang hampir sama (BMKG, 2019). Sebagai contoh citra satelit Himawari-8 WV *Enhanced* ditunjukkan pada Gambar 2.3.



Gambar 2.3 Citra Himawari-8 WV Enhanced

2.3.4. Himawari-8 Rainfall Potential (RP)

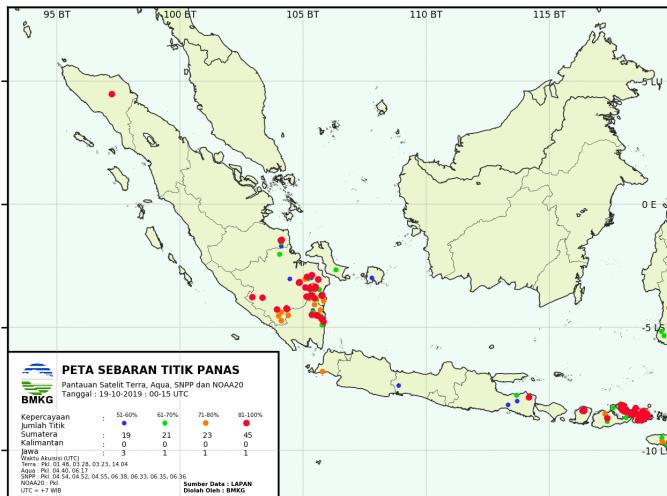
Citra Himawari-8 *Rainfall Potential* (RP) merupakan citra yang menggambarkan potensi turunnya hujan. Citra RP memiliki tiga indeks. Indeks biru menunjukkan hujan dengan intensitas rendah, warna hijau untuk hujan dengan intensitas menengah, warna kuning untuk hujan lebat, dan merah untuk hujan dengan intensitas sangat tinggi. Citra RP didapatkan dengan melihat suhu puncak awan dengan curah hujan yang dihasilkan. Citra ini hanya melihat potensi turun hujan tanpa melihat kondisi pertumbuhan awan yang ada di daerah sekitarnya. Kondisi tersebut menjadikan citra RP kurang tepat jika digunakan untuk pengamatan kondisi awan di lingkungan yang diinginkan (BMKG, 2019). Contoh citra Himawari-8 *Rainfall Potential* (RP) disajikan pada Gambar 2.4.



Gambar 2.4 Citra Himawari-8 Rainfall Potential (RP)

2.3.5. Citra *Hotspot* MODIS

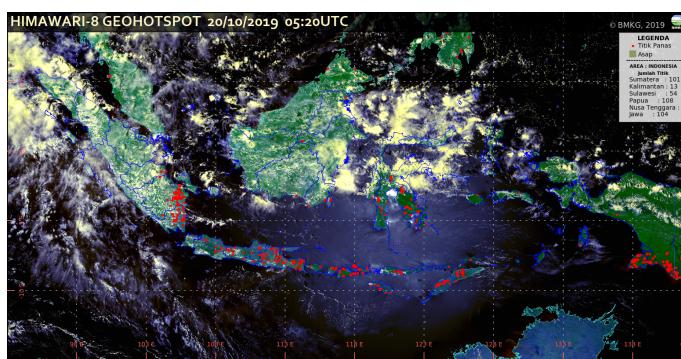
Citra *Hotspot* MODIS merupakan citra yang digunakan untuk melihat keberadaan titik api. Citra ini diperoleh dari penangkapan sensor MODIS oleh dua satelit yaitu satelit terra, dan aqua. Hasil keluaran dari pengambilan citra modis memberi gambaran titik-titik yang sedang mengalami kebakaran hutan. Pembacaan titik kebakaran hutan dilihat dari anomali suhu panas yang terawasi. Pengawasan yang dibaca oleh satelit yaitu persatu kilometer persegi. Pengambilan citra ini dilakukan 2 sampai 4 kali sehari. Pengambilan citra MODIS menjadi terganggu jika wilayah yang dipantau sedang tertutup oleh awan (BMKG, 2019). Contoh hasil dari pengambilan citra MODIS ditunjukkan pada Gambar 2.5.



Gambar 2.5 Citra Hotspot MODIS

2.3.6. Citra Himawari-8 *GeoHotspot*

Citra Himawari-8 *GeoHotspot* merupakan citra satelit yang digunakan untuk mengamati potensi terjadinya kebakaran hutan. Potensi kebakaran dilihat dari anomali suhu yang terjadi di wilayah tertentu. Berbeda dengan citra modis, citra ini diambil dengan infra merah untuk *filtering* awan. Citra *GeoHotspot* juga dapat mengamati sebaran asap yang ditandai dengan warna coklat. Titik kebakaran ditandai dengan titik berwarna merah (BMKG, 2019). Sebagai contoh pengambilan citra Himawari-8 *GeoHotspot* ditunjukkan pada Gambar 2.6.



Gambar 2.6 Citra Himawari-8 *GeoHotspot*

2.4. Awan

Awan merupakan sebuah kumpulan dari titik uap air. Kumpulan uap air terbentuk dari hasil proses daur air atau biasa disebut dengan daur hidrologi. Sebuah awan tersusun dari butiran air yang membeku, dan bergantung pada atmosfer bumi. Kumpulan dari butiran air membentuk sebuah awan yang bermacam-macam. Setiap awan memiliki bentuk, ukuran, dan ketinggian yang berbeda-beda. Jika dilihat dari ketinggiannya, awan dibedakan menjadi 3 jenis. Jenis pertama yaitu awan tinggi. Awan tinggi memiliki ketinggian 6-12 kilometer di atas permukaan laut. Awan yang termasuk dalam golongan awan tinggi diantaranya yaitu awan cirrus, awan cirromulus, dan awan cirrostratus. Jenis awan kedua yaitu awan menengah dengan ketinggian 2-6 kilometer di atas permukaan laut. Beberapa jenis awan yang termasuk dalam awan menengah yaitu altocumulus, dan altostratus. Jenis awan yang terakhir yaitu awan rendah. Ketinggian yang dimiliki oleh awan rendah kurang dari 2 kilometer di atas permukaan laut. Awan yang masuk dalam awan rendah yaitu awan cumulus, stratocumulus, stratus, nimbostratus, dan cumulonimbus (Utoyo, 2007). Berikut penjelasan untuk tiap tiap awan

1. Awan Cirrus

Awan cirrus merupakan awan yang terbentuk dari uap air kemudian menjadi uap air. Bentuk dari awan cirrus seperti serat putih yang halus. Sisi dari awan cirrus tidak begitu jelas atau samar samar. Awan ini bukan termasuk dalam golongan awan hujan. Awan cirrus berada pada ketinggian 6-18 kilometer (Massinai, 2005).

2. Awan Cirrocumulus (CiCu)

Awan CiCu merupakan awan yang memiliki bentuk bulatan-bulatan kecil. Bentuk dari awan ini sering disamakan dengan bentuk ombak yang berada pada pasir pantai. Keberadaan awan ini sangat singkat. Ketinggian awan CiCu mencapai 6-12 kilometer (Massinai, 2005).

3. Awan Cirrostratus (CiSt)

Awan Cirrostratus memiliki bentuk yang halus, dan menutup seluruh permukaan langit. Bentuk dari awan ini seperti anyaman yang tidak teratur. Awan CiSt merupakan awan yang dapat menimbulkan efek halo jika berada di tempat dengan iklim tropis. Awan CiSt dapat memicu hujan gerimis disaat hari sedang panas (Massinai, 2005).

4. Awan Alto Cumulus (ACu)

Awan Alto Cumulus (ACu) memiliki bentuk seperti bola kapas yang bergandengan. Awan ACu terjadi pada saat massa udara stabil, dan kering tertiu oleh angin kemudian berkumpul pada satu titik. Awan ACu merupakan salah satu awan yang dapat menimbulkan hujan (Nugraheny, 2015).

5. Awan Alto Stratus (ASt)

Awan Alto Stratus (ASt) merupakan awan dengan bentuk yang melebar. Lebar yang dimiliki oleh awan ini dapat menutupi seluruh bagian langit yang tampak dari bumi. Awan ASt terbentuk pada saat tertentu. Terbentuknya awan ASt hanya terbentuk pada sore hingga matahari terbit. Awan ASt merupakan awan yang membawa potensi hujan ringan (Nugraheny, 2015).

6. Awan Stratocumulus (StCu)

Awan Stratocumulus (StCu) merupakan awan dengan ketinggian 1.5 hingga 2 kilometer di atas permukaan laut. Bentuk awan StCu seperti bola dengan

sebaran yang sangat luas. Keberadaan awan ini dapat ditemui di daerah pesisir. Keberadaan awan StCu dapat berpotensi hujan lokal (Nugraheny, 2015).

7. Awan Stratus (St)

Awan Stratus (St) merupakan awan dengan bentuk seperti serat putih yang berlapis-lapis. Ketinggian awan ini mencapai 2 kilometer di atas permukaan laut. Potensi hujan yang diakibatkan oleh awan St yaitu hujan ringan (Nugraheny, 2015).

8. Awan Nimbo Stratus (NiSt)

Awan Nimbo Stratus (NiSt) memiliki bentuk yang tidak pasti. Penutupan awan NiSt dapat menutupi wilayah yang sangat luas. Ketinggian yang dimiliki oleh awan ini yaitu 0.6 hingga 3 kilometer di atas permukaan laut. Potensi hujan yang dimiliki awan ini yaitu hujan ringan hingga sedang dengan durasi waktu lama (Nugraheny, 2015).

9. Awan Cumulus

Awan Cumulus merupakan awan yang tumbuh secara vertikal. Ketinggian awan cumulus mencapai 1 kilometer di atas permukaan laut. Awan cumulus memiliki bentuk seperti bunga kol dengan warna putih cerah. Munculnya awan cumulus pada musim kemarau atau saat langit dalam keadaan cerah (Nugraheny, 2015).

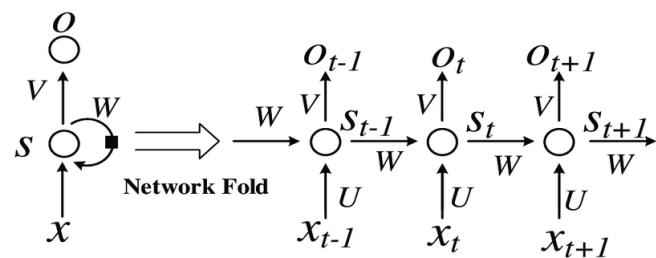
10. Awan Cumolonimbus (Cb)

Awan Cumolonimbus (Cb) merupakan awan dengan volume yang besar. Ketinggian awan Cb berada pada ketinggian 2 kilometer di atas permukaan laut. Ketebalan awan Cb dapat mencapai 16 kilometer di atas permukaan laut. Awan Cb merupakan awan yang padat, dan terdapat badai petir di dalamnya. Awan Cb merupakan awan yang tumbuh secara vertikal. Awan Cb memiliki warna

putih pada bagian atas, dan kelabu gelap pada bagian bawah. Awan Cb memiliki potensi hujan lebat hingga badai disertai petir (PaskiPermana, Sepriando, & Pertiwi, 2017).

2.5. Recurrent Neural Networks (RNNs)

Recurrent Neural Networks merupakan salah satu metode yang ada pada jaringan syaraf buatan. *Recurrent Neural Networks* merupakan jaringan yang diproses secara berulang-ulang. Jika dibandingkan dengan metode *feedforward*, *perceptron*, dan CNN, maka jaringan RNNs merupakan jaringan yang lebih baik digunakan untuk data yang bersifat sekuensial. Bentuk sekuensial merupakan sebuah keterkaitan antara data masukan pada data pertama dengan data masukan pada data selanjutnya. Keunggulan yang dimiliki RNNs pada data sekuen yaitu memori yang disimpan dari masukan pertama hingga tahap terakhir. Penyimpanan memori pada tahap sebelumnya, menyebabkan keluaran yang dihasilkan pada langkah t tidak hanya bergantung pada masukan pada langkah t , melainkan bergantung pada memori yang tersimpan pada tahap sebelumnya (Mikolov, Karafiat, Burget, Cernocky, & Khudanpur, 2010). Proses yang berjalan dalam RNNs digambarkan pada Gambar 2.7.



Gambar 2.7 Struktur Recurrent Neural Networks (RNNs)

Keterangan :

x = Variabel masukan dengan $x \in \mathbb{R}^{I \times 1}$

I = Panjang variabel masukan

U = Bobot antara masukan, dan *hidden layer*, dengan $U \in \mathbb{R}^{H \times 1}$

H = Banyak Pengulangan pada RNNs

s = Memori yang tersimpan, dengan $s \in \mathbb{R}^{H \times 1}$

W = Bobot antar memori pada RNNs, dengan $W \in \mathbb{R}^{H \times H}$

V = Bobot antara *Hidden* menuju keluaran, dengan $V \in \mathbb{R}^{K \times H}$

K = Panjang variabel keluaran

o = Variabel masukan dengan $o \in \mathbb{R}^{K \times 1}$

Melihat dari Gambar 2.7 terdapat beberapa elemen yang harus dipahami dalam sebuah proses RNNs. Nilai masukan pada RNNs disimbolkan dengan x_t yang menyatakan data x pada waktu t . Nilai s_t merupakan memori yang disimpan oleh RNNs pada waktu t . Nilai s_t didapatkan dengan mengambil nilai masukan, dan memori s_{t-1} yang masing-masing dikalikan dengan bobot U , dan W . Fungsi aktivasi yang dapat digunakan untuk memperoleh nilai s_t yaitu tangen hiperbolik (Zaremba, Sutskever, & Vinyals, 2014). Fungsi s_t ditunjukkan pada Persamaan 2.2.

$$s_t = \tanh(U.x_t + W.s_{t-1}) \quad (2.2)$$

Setelah mendapatkan nilai memori yang baru, maka dapat dilanjutkan menghitung nilai keluaran dari nilai s_t , dan bobot V menggunakan fungsi *softmax* (Zaremba, Sutskever, & Vinyals, 2014). Perhitungan keluaran dengan fungsi *softmax* ditunjukkan pada Persamaan 2.3.

$$o_t = softmax(V \cdot s_t) \quad (2.3)$$

$$= \frac{e^{V \cdot s_t}}{\sum_{t=1}^T e^{V \cdot s_t}}$$

Nilai yang dihasilkan o_t , merupakan nilai prediksi yang dihasilkan oleh RNNs. Hasil dari perhitungan o_t , selanjutnya dihitung nilai *loss function*. Perhitungan nilai *loss function* ditujukan untuk mengetahui tingkat keberhasilan sistem dalam pengenalan sebuah pola. Nilai *loss function* didapat dengan menggunakan Persamaan 2.4. Sedangkan untuk menghitung nilai *loss function* untuk setiap langkahnya menggunakan Persamaan 2.5, dengan nilai y_t merupakan nilai keluaran sebenarnya (Zaremba, Sutskever, & Vinyals, 2014).

$$L(o, y) = -\frac{1}{T} \sum_{t=1}^T y_t \log o_t \quad (2.4)$$

$$L_t(o_t, y_t) = -y_t \log o_t \quad (2.5)$$

Kesalahan yang didapatkan dari *loss function* dapat dijadikan acuan untuk memperbaiki proses yang sedang berjalan. Perbaikan dilakukan dengan memperbarui nilai bobot di setiap tahapnya. Algoritma RNNs, melakukan perbaikan nilai bobot menggunakan proses *Back Propagation Through Time* (BPTT). Proses BPTT digunakan untuk menghitung eror yang saling berkaitan antara eror pada saat t , $t - 1$, hingga $t - n$. Langkah pertama untuk memperbaiki bobot yaitu menghitung nilai δ untuk setiap neuronnya. Pada neuron keluaran, nilai δ^y dihitung menggunakan Persamaan 2.6. Sedangkan pada lapisan memori nilai δ^s diperoleh menggunakan Persamaan 2.7. Kecuali pada tahap paling kanan RNNs, nilai $\delta_{i,T+1}^s = 0$ (Gregor, Danihelka, Graves, Rezende, & Wierstra, 2015).

$$\delta_{i,t}^y = \frac{\partial L_t}{\partial y_{i,t}} \quad (2.6)$$

$$\delta_{i,t}^s = \left(\sum_{j=1}^K V_{i,j} \cdot \delta_{j,t}^y + \sum_{n=1}^H W_{i,n} \cdot \delta_{n,t+1}^h \right) \cdot f'(s_{i,t}) \quad (2.7)$$

Setelah mendapatkan nilai δ , dilanjutkan untuk menghitung nilai perbaikan bobot pada setiap lapisanya. Nilai perbaikan $\triangle V$, $\triangle W$, $\triangle U$ masing-masing dihitung menggunakan Persamaan 2.8, Persamaan 2.9, dan Persamaan 2.10.

$$\triangle V i, j = \sum_{t=1}^T \delta_{j,t}^y \cdot s_{i,t} \quad (2.8)$$

$$\triangle W_{i,j} = \sum_{t=1}^T \delta_{j,t}^h \cdot s_{i,t-1} \quad (2.9)$$

$$\triangle U i, j = \sum_{t=1}^T \delta_{j,t}^h \cdot x_{i,t} \quad (2.10)$$

Nilai perbaikan pada masing-masing bobot yang sudah didapat, selanjutnya dijumlahkan dengan bobot yang lama untuk mendapatkan pembaruan bobot yang baru. Perhitungan bobot yang baru ditunjukkan pada Persamaan 2.11, sampai dengan Persamaan 2.13 (Gregor, Danihelka, Graves, Rezende, & Wierstra, 2015).

$$V = V + \Delta V \quad (2.11)$$

$$W = W + \Delta W \quad (2.12)$$

$$U = U + \Delta U \quad (2.13)$$

2.6. Reservoir Computing (RC)

Reservoir Computing (RC) merupakan salah satu teknik yang digunakan dalam *machine learning*. *Reservoir computing* memetakan data masukan menuju dimensi yang lebih tinggi. Metode *Reservoir Computing* diperkenalkan oleh Herbert Jaeger, dan Wolfgang Maass pada awal tahun 2000-an. Metode ini dikembangkan untuk meningkatkan kinerja *machine learning* dalam pengenalan

sinyal *real time*. RC sangat efisien jika digunakan untuk data-data yang bergantung pada waktu atau berkaitan antara data satu dengan data selanjutnya. Model jaringan yang dibentuk oleh *Reservoir Computing* menggunakan model pada RNNs (Schrauwen, Verstraeten, & Van Campenhout, 2007).

Terdapat tiga lapisan utama yang berada pada jaringan *Reservoir Computing*. Lapisan yang pertama yaitu lapisan masukan. Lapisan masukan pada RC berupa vektor yang memiliki panjang sesuai dengan banyaknya nilai masukan yang digunakan. Lapisan kedua yaitu lapisan tersembunyi yang biasa disebut dengan *reservoir*. Pada lapisan ini terdiri dari beberapa neuron yang saling berkaitan satu dengan yang lain. Neuron satu dengan lainnya terhubung secara acak, dan ditentukan pada proses awal pembentukan jaringan. Lapisan pertama terhubung dengan lapisan kedua menggunakan bobot sebagai faktor pengali nilai masukan menuju lapisan *reservoir*. Hubungan yang dibentuk antara masukan, dan *reservoir* ditentukan di awal proses. Pada tahap selanjutnya, hubungan antara lapisan masukan, dan *reservoir* tidak mengalami pembelajaran (Van der Sande, Brunner, & Soriano, 2017). Selain memiliki hubungan dengan lapisan masukan, neuron pada *reservoir* juga terhubung dengan neuron lain yang ada dalam *reservoir*. Hubungan antar neuron dalam *reservoir* terjadi secara acak. Hubungan antar neuron berlangsung secara berulang, dan menggunakan konsep dari *Recurrent Neural Networks* (RNNs). Penggunaan konsep RNNs membuat *reservoir* mampu memiliki memori untuk menyimpan informasi masukan yang bersifat sekuen (Klibisz, 2016).

Lapisan terakhir pada RC yaitu lapisan keluaran. Lapisan keluaran merupakan sebuah vektor dengan ukuran sesuai dengan banyak variabel keluaran yang diinginkan. Diantara lapisan keluaran, dan *reservoir* dihubungkan dengan

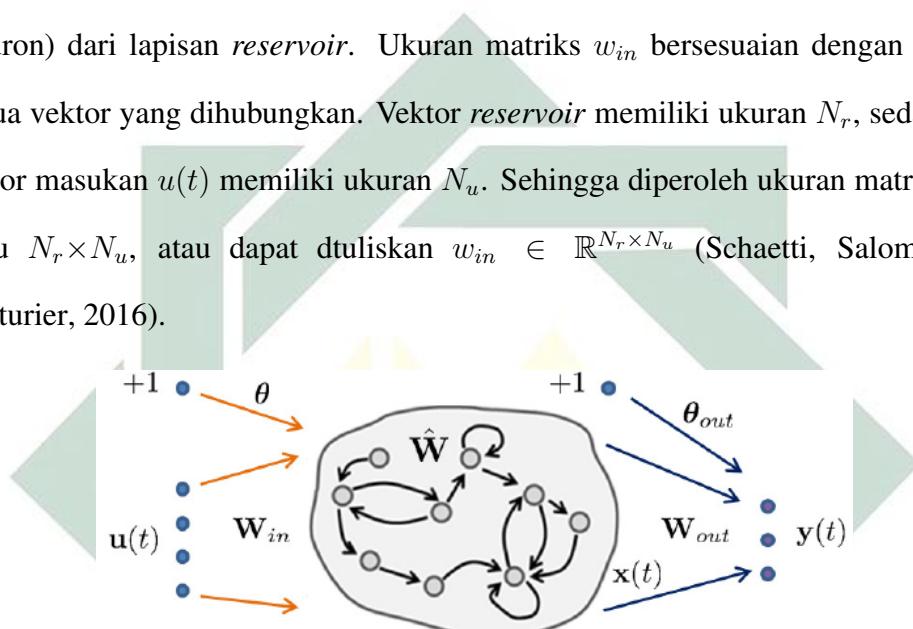
sebuah nilai bobot. Berbeda dengan bobot antara lapisan masukan, dan *reservoir*, bobot pada lapisan ini mengalami pembelajaran. Proses pembelajaran dilakukan dengan perhitungan mundur. Proses pembelajaran dalam RC menggunakan proses seperti pada *backpropagation*. Nilai bobot akan diperbaiki dengan menghitung secara berulang pada lapisan *reservoir*, dan lapisan keluaran. Perbaikan bobot yang terjadi hanya di lapisan keluaran, menjadikan jaringan ini memiliki kecepatan yang lebih tinggi jika dibandingkan dengan jaringan sebelumnya. Beberapa metode yang menggunakan pengembangan dari RC yaitu *Liquid State Machine*, *Echo State Network*, dan *Backpropagation-Decorrelation learning rule* (Klibisz, 2016).

2.7. Echo State Network (ESN)

Echo state network merupakan salah satu jenis jaringan dari *reservoir computing*. Model *Echo state network* merupakan pengembangan dari RNN. Sebuah jaringan ESN merupakan salah satu jaringan RNNs yang berjalan secara efisien. Seperti sistem yang bekerja dalam jaringan RNN, sistem yang berjalan dalam jaringan ESN merupakan perhitungan dinamis yang bergantung pada waktu. Perhitungan yang terbentuk dalam ESN berjalan dalam sebuah *reservoir* secara berulang, dan tidak terlatih. Berjalannya jaringan ESN secara berulang menjadikan ESN hampir sama dengan konsep dari markovian. Konsep yang dibentuk oleh markovian yaitu menggunakan masukan di waktu sebelumnya menjadi acuan dalam proses di waktu mendatang. Perhitungan secara markovian, menciptakan pemikiran baru dalam pengembangan jaringan ESN (Schaetti, Salomon, & Couturier, 2016).

Salah satu bentuk pengembangan dari jaringan ESN yaitu *Leaky Integrator ESN* (LI-ESN). Sebagai penggambaran dari jaringan LI-ESN disajikan pada

Gambar 2.8. Jaringan LI-ESN, dalam Gambar 2.8, menunjukkan jaringan dengan masukan berupa vektor $u(t)$ (dalam waktu t). Vektor masukan $u(t)$ merupakan elemen dari R^{N_u} . Vektor masukan $u(t)$, kemudian dipetakan pada lapisan *reservoir* dengan sebuah bobot W_{in} , dan penambahan bias θ . Bobot w_{in} , merupakan bobot yang menghubungkan setiap elemen $u(t)$ tepat satu elemen (neuron) dari lapisan *reservoir*. Ukuran matriks w_{in} bersesuaian dengan ukuran kedua vektor yang dihubungkan. Vektor *reservoir* memiliki ukuran N_r , sedangkan vektor masukan $u(t)$ memiliki ukuran N_u . Sehingga diperoleh ukuran matriks w_{in} yaitu $N_r \times N_u$, atau dapat dituliskan $w_{in} \in \mathbb{R}^{N_r \times N_u}$ (Schaetti, Salomon, & Couturier, 2016).



Gambar 2.8 Struktur Jaringan *Echo State Network* (ESN)

Hasil pemetaan dari vektor masukan $u(t)$ dengan w_{in} merupakan vektor *reservoir* $x(t)$. Berdasarkan kinerja dari jaringan berulang RNN, nilai vektor $x(t)$ juga dipengaruhi oleh setiap neuron yang ada di dalamnya. Sebagai penghubung antara neuron dalam *reservoir* yaitu nilai bobot w_x . Setiap neuron dalam *reservoir* terhubung satu sama lain hingga memiliki keterkaitan sebanyak jumlah neuron atau N_r . Keterkaitan semua *neuron reservoir* mengakibatkan ukuran matriks bobot w_x adalah $N_r \times N_r$, atau dinotasikan sebagai $w_x \in \mathbb{R}^{N_r \times N_r}$. Nilai setiap neuron *reservoir* ($x(t)$) didapatkan menggunakan Persamaan 2.14 (Schaetti, Salomon, &

Couturier, 2016).

$$x(t) = (1 - \alpha)x(t - 1) + \alpha \tanh(w_{in}u(t) + w_\theta\theta + w_xx(t - 1)) \quad (2.14)$$

Melihat dari Persamaan 2.14, Nilai $x(t)$ dipengaruhi oleh nilai resevoir di waktu sebelumnya atau $x(t - 1)$. Nilai pada lapisan *reservoir* dipengaruhi juga oleh θ yang diperoleh dari distribusi normal. Distribusi normal θ memiliki interval $[-skala_{in}, skala_{in}]$, dengan $skala_{in}$ merupakan skala dari nilai masukan. Pada Persamaan 2.14, terdapat nilai parameter *leaky* (α) yang mempunyai interval $[0, 1]$. Berbeda dengan parameter yang lain, nilai matriks w_x merupakan nilai yang dibangkitkan secara acak menggunakan distribusi normal antara nol sampai satu. Pada tahap selanjutnya nilai w_x diperbaiki menggunakan Persamaan 2.15, dengan I merupakan matriks identitas (Schaetti, Salomon, & Couturier, 2016).

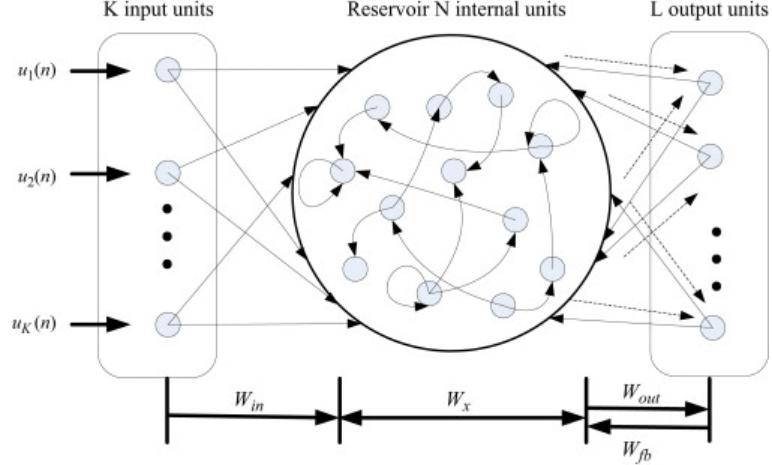
$$w_x = (1 - \alpha)I + \alpha w_x. \quad (2.15)$$

Setelah menyelesaikan perhitungan pada lapisan *reservoir*, dilanjutkan menghitung keluaran pada lapisan *readout*. Nilai keluaran ($y(t)$) merupakan sebuah vektor yang memiliki panjang N_y . Penghubung antara neuron dalam *reservoir*, dan lapisan *readout* adalah bobot keluaran (w_{out}). Setiap neuron dalam *reservoir* terhubung tepat satu pada lapisan keluaran. Keterhubungan antara *reservoir*, dan keluaran menghasilkan ukuran matriks bobot w_{out} yaitu $N_y \times N_r$, atau dapat dinotasikan dengan $w_{out} \in \mathbb{R}^{N_y \times N_r}$. Nilai pada lapisan *readout* didapatkan dengan menggunakan Persamaan 2.16 (Koryakin & Butz, 2012).

$$y(t) = w_{out}x(t) + \theta_{out} \quad (2.16)$$

Hasil yang diperoleh disimpan untuk menjadi perbaikan pada proses di waktu yang akan datang. Sedikit berbeda dengan proses pada jaringan syaraf lainnya, proses pembelajaran yang dimiliki ESN terdapat pada lapisan terakhir. bobot w_{in} , w_x , dan w_{fb} tidak mengalami pembelajaran. Nilai bobot w_{back} merupakan bobot yang menghubungkan antara *readout* kembali ke lapisan *reservoir*. Jumlah keterkaitan antara w_{fb} , dan w_{out} adalah sama, tetapi memiliki arah yang berbeda. Sehingga matriks w_{fb} memiliki ukuran berkebalikan (*transpose*), dari ukuran matriks w_{out} . Penyampaian nilai *readout* menuju *reservoir* ditujukan untuk penyimpanan memori pada perhitungan $x(t)$ di waktu mendatang. Perhitungan nilai $x(t)$ di waktu yang akan datang diperoleh menggunakan Persamaan 2.17. Sebagai gambaran dari proses perhitungan *readout* menuju *reservoir* diilustrasikan pada Gambar 2.9 (Koryakin & Butz, 2012).

$$x(t) = (1 - \alpha)x(t-1) + \alpha \tanh(w_{in}u(t) + \theta + w_x x(t-1) + w_{fb}y(t-1)) \quad (2.17)$$



Gambar 2.9 Struktur *Echo State Network* (ESN) Menggunakan Bobot fb

Proses selanjutnya yaitu proses pembelajaran yang terjadi pada matriks w_{out} . Terdapat dua cara untuk memperbaiki nilai bobot selama proses pelatihan.

Cara pertama yaitu perhitungan menggunakan nilai $x(t)$ yang disimpan dari $t = 0$ hingga $t = T$ atau disebut dengan *offline learning*. Cara yang kedua yaitu memperbaiki bobot setiap perulangan berlangsung atau *online learning*. Proses perhitungan *offline learning* diawali dengan menghitung nilai $E(w_{out})$. Nilai $E(w_{out})$ merupakan nilai eror $y(t)$ terhadap nilai target sebenarnya ($Y(t)$), mulai dari $t = 0$ hingga $t = T$. Perhitungan nilai $E(w_{out})$ disajikan pada Persamaan 2.18. Selanjutnya, menggunakan metode *Least Square Estimation* (LSE) proses perbaikan bobot dilakukan dengan menggabungkan semua nilai $x(t)$, dan berlaku juga untuk nilai target $y(t)$. Diperoleh matriks X adalah kumpulan dari vektor $x(t)$, dan X' merupakan matriks *transpose* dari X . Diperoleh juga matriks Y adalah matriks target dari $t = 0$ hingga $t = T$. Kedua matriks yang dihasilkan digunakan untuk memperbarui nilai w_{out} menggunakan Persamaan 2.19 (Koryakin & Butz, 2012).

$$E(w_{out}) = \frac{1}{T} \sum_{t=1}^T |y(t) - Y(t)|^2 \quad (2.18)$$

$$w_{out} = (X' \times X)^{-1} X' \times Y \quad (2.19)$$

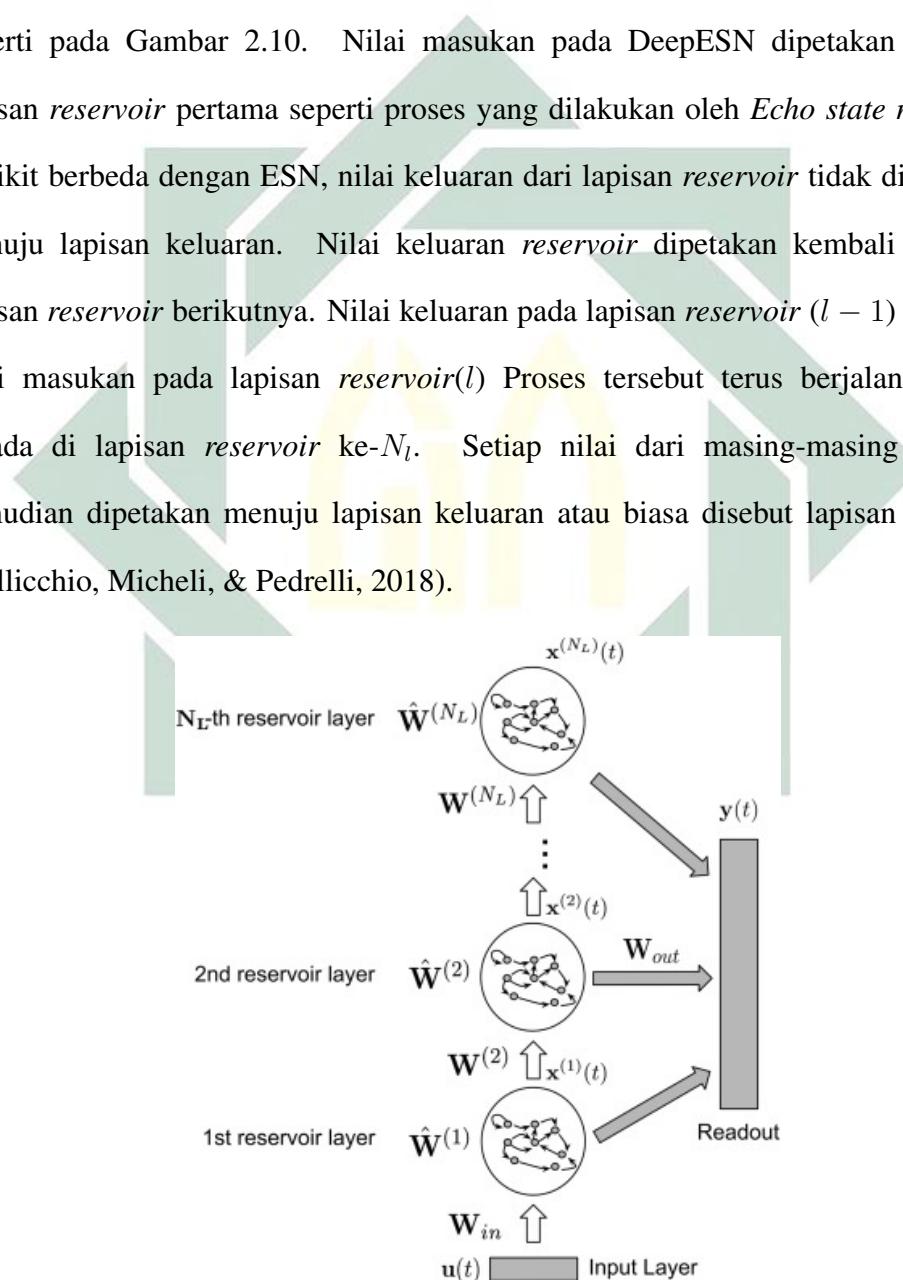
Cara kedua yaitu menggunakan *online learning*. Cara ini menggunakan perhitungan disetiap perulangannya. Metode yang digunakan untuk mengurangi eror sama seperti pada *single-layer perceptron*. Dengan nilai *learning rate* (η) merupakan suatu konstanta yang didefinisikan diawal. Nilai η memiliki rentang antara nol sampai satu. Perbaikan bobot diawali dengan menghitung Δw menggunakan Persamaan 2.20. Setelah didapatkan nilai pembaruan Δw , dihitung nilai w_{out} yang baru menggunakan Persamaan 2.21 (Yusoff & Jin, 2014).

$$\Delta w = \eta(y(t) - Y(t))x(t) \quad (2.20)$$

$$w_{out} = w_{out} + \Delta w \quad (2.21)$$

2.8. Deep Echo State Networks (DeepESN)

Deep Echo State Networks (DeepESN) merupakan suatu jaringan RC yang terdiri dari beberapa lapisan *reservoir*. Lapisan *reservoir* ditumpuk secara lurus seperti pada Gambar 2.10. Nilai masukan pada DeepESN dipetakan menuju lapisan *reservoir* pertama seperti proses yang dilakukan oleh *Echo state network*. Sedikit berbeda dengan ESN, nilai keluaran dari lapisan *reservoir* tidak diarahkan menuju lapisan keluaran. Nilai keluaran *reservoir* dipetakan kembali menuju lapisan *reservoir* berikutnya. Nilai keluaran pada lapisan *reservoir* ($l - 1$) menjadi nilai masukan pada lapisan *reservoir*(l). Proses tersebut terus berjalan hingga berada di lapisan *reservoir* ke- N_l . Setiap nilai dari masing-masing lapisan kemudian dipetakan menuju lapisan keluaran atau biasa disebut lapisan *readout* (Gallicchio, Micheli, & Pedrelli, 2018).



Gambar 2.10 Struktur Deep Echo State Network (DeepESN)

Mengacu pada Persamaan 2.14, proses transisi pada setiap lapisnya dihitung menggunakan Persamaan 2.22.

$$x^{(l)}(t) = (1 - \alpha^{(l)})x^{(l)}(t-1) + \alpha^{(l)}f(w_{in}^{(l)}i^{(l)}(t) + w_{\theta}^{(l)}\theta^{(l)} + w_x^{(l)}x^{(l)}(t-1)) \quad (2.22)$$

Dimana l menandakan posisi layer *reservoir*. Nilai l dimulai dari satu sampai L yang menandakan posisi *reservoir* terakhir. Pada Persamaan 2.22 menggunakan fungsi aktivasi tangen hiperbolik yang disimbolkan dengan f . Parameter lain yang mempengaruhi $x(t)$ yaitu $w_x^{(l)}$ yang merupakan bobot penghubung antar neuron yang ada pada lapisan *reservoir* ke- l , nilai $\theta^{(l)}$ adalah bias pada lapisan ke- l , dan nilai $\alpha^{(l)}$ sebagai parameter *leaky* pada lapisan *reservoir* ke- l . Berbeda dengan persamaan yang dimiliki oleh ESN, keluaran *reservoir* pada DeepESN memiliki ukuran $w_{in}^{(l)}$ yang berbeda-beda. Matriks $w_{in}^{(l)}$ berukuran $N_r \times N_u$ jika nilai $l = 1$. Jika nilai $l > 1$ maka ukuran matriks $w_{in}^{(l)}$ adalah $N_r \times N_r$. Perbedaan dengan persamaan ESN terdapat pada vektor masukan (, Gallicchio & Micheli(2017).

Pada Persamaan 2.22 memiliki vektor masukan yaitu i . Nilai vektor i memiliki dua kondisi, jika $l = 1$ maka nilai i merupakan vektor masukan. Kondisi kedua yaitu ketika $l > 1$, maka nilai i mengambil nilai $x(t)$ pada layer sebelumnya. Ketika semua nilai keluaran disetiap *reservoir* sudah didapatkan, selanjutnya dapat dihitung vektor pada lapisan *readout*. Vektor *readout* didapatkan dengan mengalikan setiap nilai keluaran *reservoir* dengan matriks bobot w_{out} , dan menambahkan bias θ_{out} . Matriks w_{out} meghubungkan vektor $x(t)$ disetiap lapisan dengan jumlah neuron pada lapisan *readout*. Sehingga didapatkan ukuran matriks w_{out} adalah $N_y \times (N_l \times N_r)$ persamaan untuk memperoleh nilai keluaran didefinisikan pada Persamaan 2.23. Pembelajaran yang digunakan pada DeepESN sama seperti pembelajaran ESN. Proses pembelajaran hanya berlangsung untuk

memperbaiki nilai w_{out} . Perbaikan w_{out} untuk menurunkan eror menggunakan cara yang sama pada Persamaan 2.19. Perhitungan untuk mendapatkan nilai w_{out} dilakukan hanya pada proses pelatihan. Pada saat pengujian proses perhitungan langsung menghitung nilai keluaran menggunakan Persamaan 2.23 (Lukocevicius & Jaeger, 2009).

$$y(t) = w_{out} \left[x^{(1)}(t)x^{(2)}(t) \cdots x^{(L)}(t) \right]^T + w_\theta \theta_{out} \quad (2.23)$$

2.9. Peak Signal-to-Noise Ratio (PSNR)

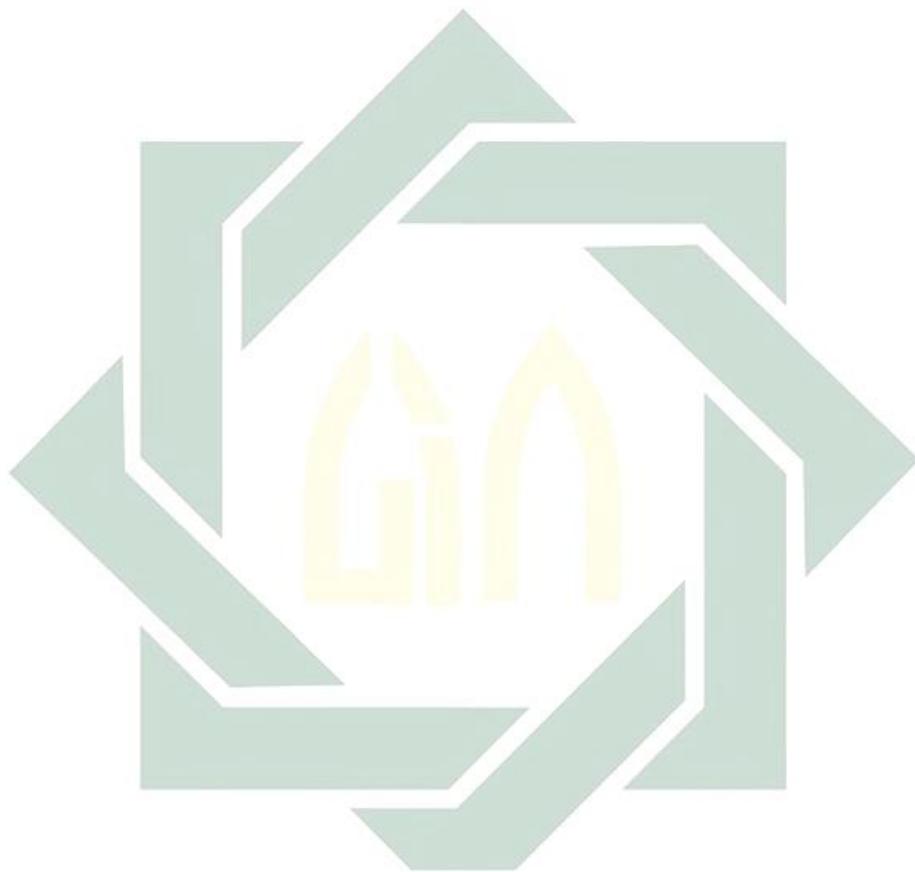
Peak Signal-to-Noise Ratio (PSNR) merupakan salah satu metode untuk melihat kemiripan citra. Nilai kemiripan disajikan dengan satuan *decibel* (db). Perhitungan kemiripan citra dilakukan dengan menghitung selisih eror pada tiap pikselnya. Metode PSNR menggunakan *Mean Square Error* (MSE) sebagai metode penghitungan eror. Semakin kecil selisih eror dari kedua citra yang dibandingkan maka semakin besar nilai kemiripan dua citra tersebut. Langkah awal untuk menghitung nilai PSNR yaitu dengan menghitung nilai MSE. Perhitungan eror MSE ditunjukkan pada Persamaan 2.24. $h_{i,j}$ nilai pada matriks citra asli pada baris ke- i , dan kolom ke- j . Sedangkan $g_{i,j}$ merupakan nilai pada matriks citra hasil prediksi pada baris ke- i , dan kolom ke- j . Nilai M , dan N menunjukkan ukuran citra. M menunjukkan banyaknya baris pada matriks citra, dan N menunjukkan banyaknya kolom (Wolf & Pinson, 2009) pada matriks citra.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (h_{i,j} - g_{i,j})^2 \quad (2.24)$$

Nilai MSE yang sudah didapatkan selanjutnya menjadi nilai masukan untuk menghitung nilai PSNR. Dengan nilai C_{max} merupakan nilai piksel maksimal pada

citra. Nilai PSNR dihitung menggunakan fungsi logaritma yang disajikan pada persamaan 2.25.

$$PSNR = 10 \log 10 \left(\frac{C_{max}^2}{MSE} \right) \quad (2.25)$$



BAB III

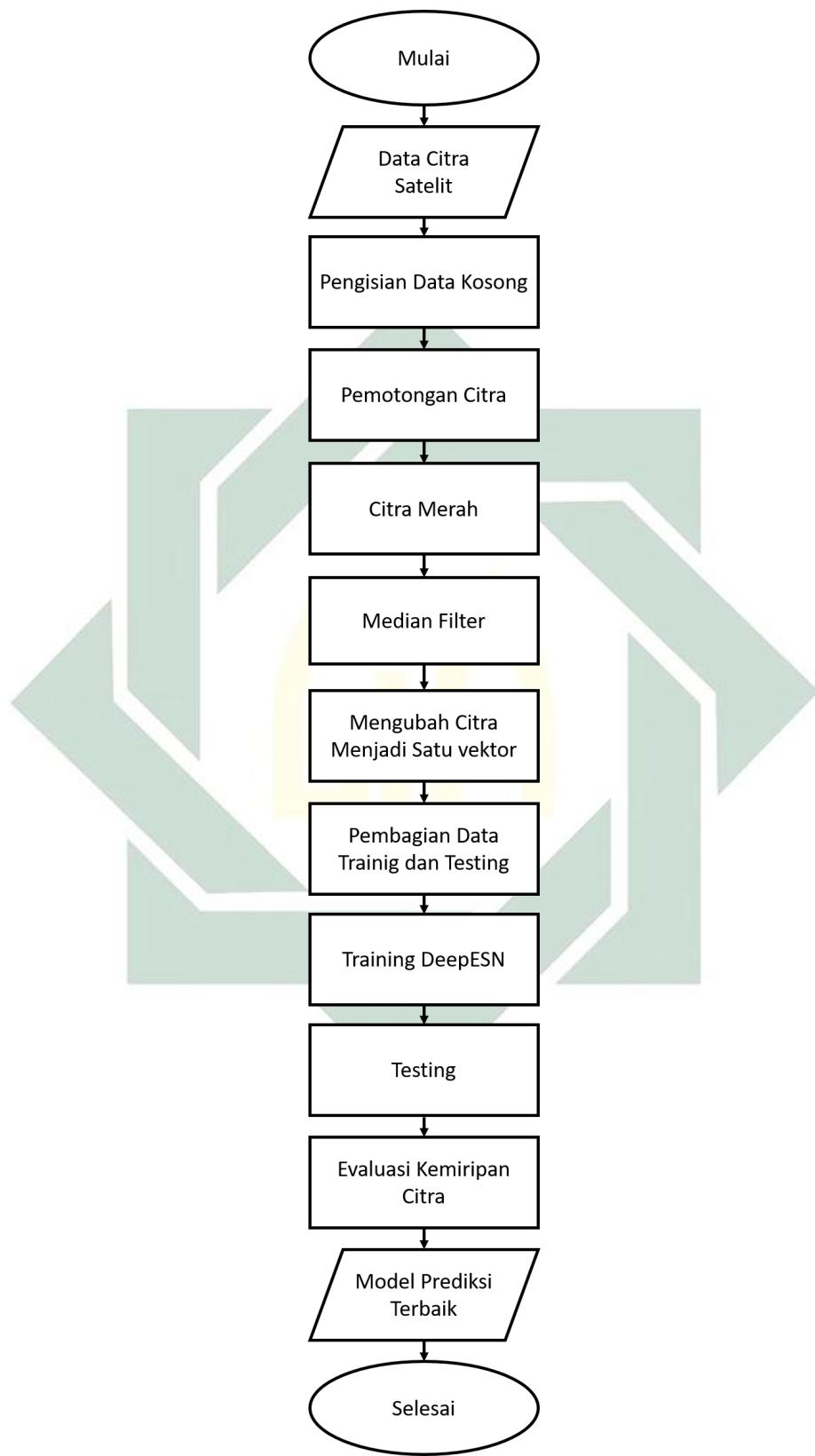
METODE PENELITIAN

3.1. Data Penelitian

Data yang digunakan merupakan data yang diambil dari satelit. Data satelit yang digunakan berupa citra yang diambil dari satelit. Citra satelit diambil dari satelit himawari yang mengawasi wilayah Indonesia. Pada proses selanjutnya mengambil citra yang difokuskan pada daerah tertentu saja. Pengambilan citra satelit dilakukan setiap satu jam sekali. Pada penelitian ini data satelit diambil dengan durasi 11 bulan, yang dimulai dari tanggal 1 Januari 2019 sampai 30 November 2019. Pengambilan jangka waktu 11 bulan ditujukan untuk menyeimbangkan data antara data pada musim hujan dengan data musim kemarau. Selain pengambilan waktu tersebut ditujukan untuk meningkatkan kinerja sistem dalam pengenalan dua pola yang berbeda. Data yang digunakan berupa citra RGB. Nilai piksel dalam citra menunjukkan suhu yang dimiliki sebuah awan. Awan dengan suhu yang rendah ditandai dengan warna yang semakin merah, sedangkan awan dengan suhu tinggi memiliki warna biru. Piksel akan bernilai 0 atau hitam jika tidak ada awan di daerah yang diamati. Nilai suhu terendah yang dapat dibaca dalam citra yaitu $-100^{\circ}C$, dan suhu maksimal $60^{\circ}C$. Setiap suhu yang dibaca dikonversi dalam nilai piksel RGB.

3.2. Rancangan Penelitian

Rancangan penelitian ini disajikan dalam diagram alir pada Gambar 3.1.



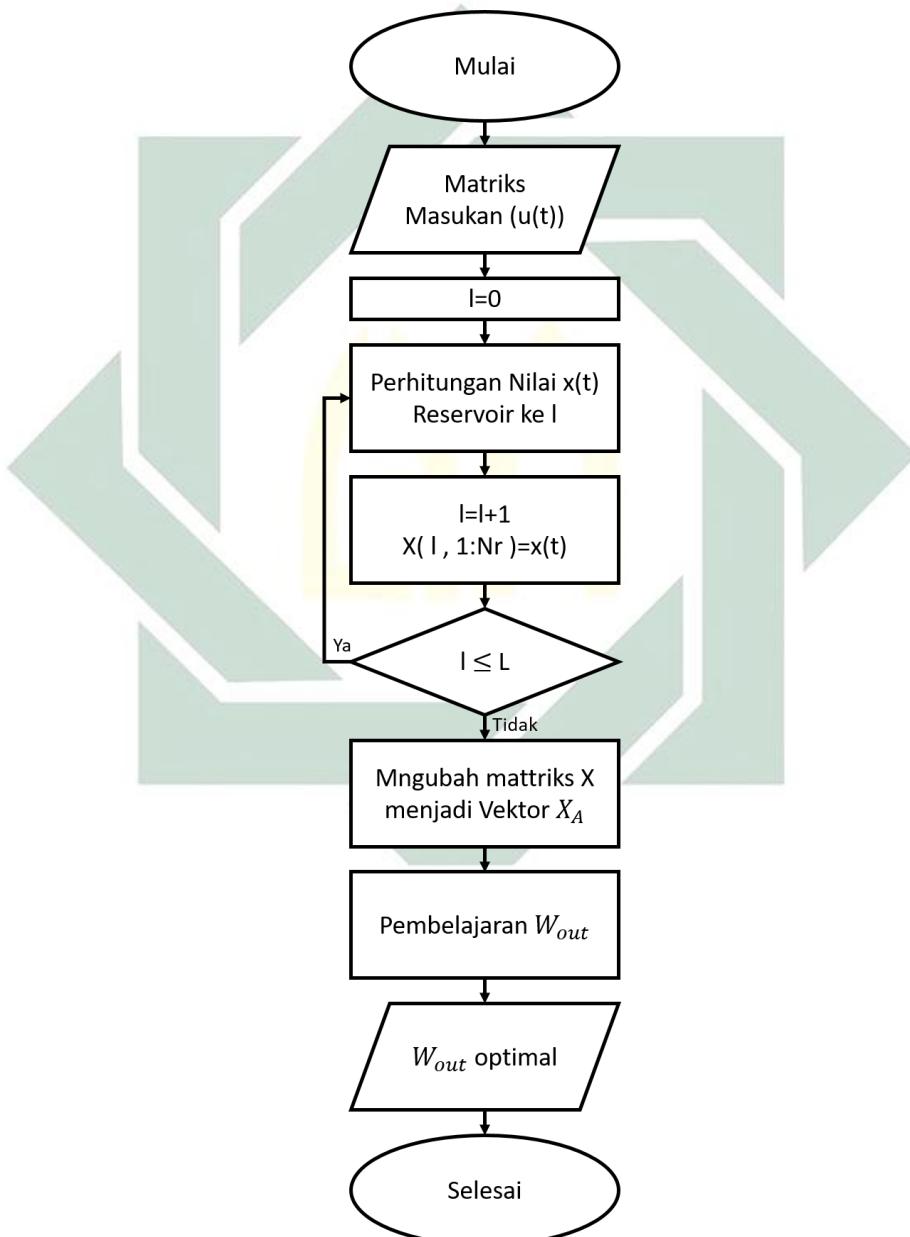
Gambar 3.1 Diagram Alir Rancangan Penelitian

Penelitian diawali dengan menggunakan data *input* berupa citra satelit. Pada data awal citra awal merupakan citra RGB. Penelitian ini hanya mengambil satu bagian dari citra RGB. Terdapat beberapa waktu satelit tidak mengambil gambar awan. Akibatnya terjadi kekosongan data dibeberapa jam. Untuk mengisi data citra yang belum diambil diambil citra dibeberapa menit terdekat dari data kosong. Setelah semua data kosong terisi citra satelit yang berupa citra RGB diambil layer merahnya. Pengambilan cira merah kerena keberadaan awan cb ditunjukkan dengan suhu yang sangat dingin atau berwarna merah. Citra satelit yang sudah ada dipotong pada wilayah tertentu. Pemotongan ditujukan untuk memperoleh citra pada wilayah yang akan diamati. Pomotongan diambil dengan mengambil ukuran piksel 50×50 piksel pada wilayah Kalimantan Selatan, dan sekitarnya. Selanjutnya semua matriks citra diubah kedalam bentuk vektor $M \times N$. M Menunjukkan jumlah piksel yang dalam citra, dan N merupakan panjang waktu dari data yang digunakan.

Proses selanjutnya membagi data menjadi data pelatihan dan data pengujian. Data pelatihan merupakan data yang diambil dari $t = 1$ hingga panjang waktu pembelajaran yang diinginkan ($t = T$). Data pengujian diambil dari waktu $T + 1$ sampai data yang terakhir. Pembagian data pelatihan ditujukan untuk memberi pembelajaran sistem prediksi. Kelompok data pengujian ditujukan untuk menguji hasil pengenalan pola pada sistem prediksi. Pengenalan pola pada proses prediksi dilakukan menggunakan metode DeepESN. Sebagai gambaran proses pengenalan pola pada DeepESN ditunjukkan pada Gambar3.2.

Matriks masukan pada DeepESN diproses satu persatu dari banyak data yang sudah ada. Selanjutnya menghitung nilai pada setiap lapisan *reservoir* menghasilkan sebuah vektor $x(t)$. Perhitungan nilai $x(t)$ didapatkan menggunakan Persamaan 2.22. Setelah mendapatkan nilai $x(t)$ pada semua *reservoir*, dilanjutkan

menghitung nilai bobot. Perhitungan nilai bobot dilakukan dengan menjadikan satu vektor dari $x(t)$ untuk semua lapisan *reservoir*. Didapatkan hasil vektor x_A dengan ukuran $Nl \times Nr$. Pada proses pembelajaran bobot menggunakan persamaan 2.19. Hasil Proses ini yaitu optimal bobot yang didapatkan dari Persamaan 2.19.



Gambar 3.2 Diagram Alir Pelatihan DeepESN

Setelah menyelesaikan proses pelatihan dilanjutkan proses pengujian.

Proses pengujian dilakukan menggunakan model yang sudah optimal. Data pengujian dimasukkan dalam proses DeepESN. Data pengujian dihitung nilai $x(t)$ menggunakan Persamaan 2.22. Proses perhitungan $x(t)$ diulang sampai pada lapisan *reservoir* terakhir. Setelah mendapatkan keluaran $x(t)$ dari masing-masing *reservoir*, nilai $x(t)$ dikumpulkan menjadi satu vektor X_A . Vektor X_A kemudian dipetakan menuju nilai keluaran menggunakan nilai bobot yang sudah optimal. Perhitungan nilai keluaran ($y(t)$) pada tahap prediksi menggunakan Persamaan 2.23.

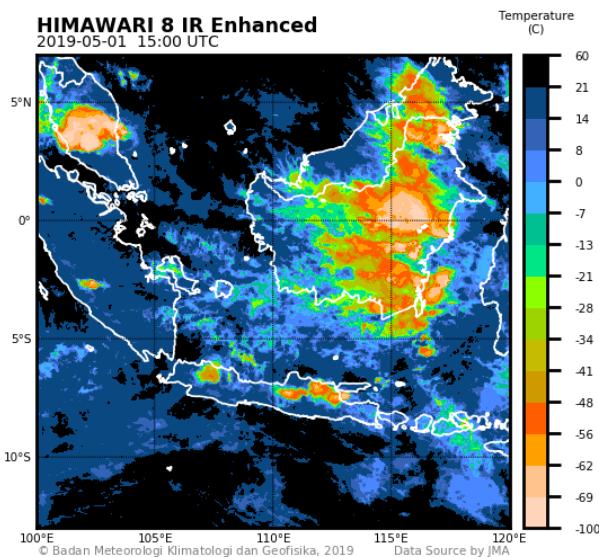
Hasil prediksi dimasukkan kedalam sistem evaluasi untuk mendapatkan nilai kesamaan citra antara citra pada data asli dengan citra hasil prediksi. Proses evaluasi menggunakan PSNR diawali dengan menghitung nilai MSE dari kedua citra menggunakan Persamaan 2.24. Selanjutnya nilai MSE dimasukkan pada perhitungan PSNR sebagai nilai indeks keberhasilan dalam memprediksi pertumbuhan awan saat satu jam kedepan. Perhitungan PSNR dihitung menggunakan Persamaan 2.25. untuk melihat struktur DeepESN yang paling optimal dilihat dari nilai PSNR yang paling tinggi. Semakin tinggi nilai PSNR yang dihasilkan maka citra prediksi semakin mendekati citra pada data aslinya.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Preprocessing Data

Penelitian ini menggunakan data dari tanggal 1 Januari 2019 sampai dengan 30 November 2019. Data yang digunakan merupakan data yang diambil setiap satu jam sekali. Total jumlah data yang digunakan yaitu sebanyak 8014 data. Sebagai contoh data yang digunakan pada penelitian ini ditunjukkan pada Gambar 4.1. Contoh data yang ditampilkan merupakan kondisi awan pada tanggal 1 Mei 2019 jam 15.00 WIB. Data yang digunakan merupakan citra satelit himawari yang mengawasi Indonesia wilayah dua. Pengamatan wilayah dua meliputi daerah Sumatra, Jawa, Nusa Tenggara, dan Kalimantan. Citra himawari yang digunakan merupakan himawari IR *Enhanced*. Citra IR *Enhanced* merupakan citra yang mengamati suhu puncak awan. Pada citra himawari membaca suhu awan dari $-100^{\circ}C$ hingga $60^{\circ}C$. Awan cb terbaca melalui citra himawari karena ketinggian awan cb yang menjulang hingga melewati ketinggian dari awan tinggi. Selain itu, suhu yang dimiliki awan cb lebih rendah jika dibandingkan dengan suhu awan tinggi. Pada penelitian ini pembacaan awan cb hanya untuk awan cb yang memiliki ketinggian lebih dari awan tinggi dan memiliki suhu dibawah -60° . Keberadaan awan cb dengan suhu dibawah $-60^{\circ}C$ ditunjukkan warna kemerahan hingga merah pekat. Semakin pekat warna merah yang ditunjukkan pada citra himawari, maka semakin dingin dan semakin berbahaya bagi aktivitas penerbangan.

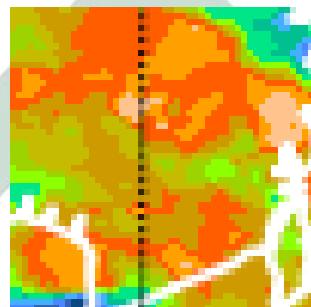


Gambar 4.1 Contoh Data Citra Himawari-8 IR Enhanced

Data yang tersedia pada satelit tidak sepenuhnya terisi. terdapat beberapa jam yang tidak diambil oleh satelit. Akibatnya, perlu dilakukan pengisian terhadap beberapa data citra yang tidak diambil oleh satelit. pada penelitian ini pengisian data kosong diberikan dengan mengambil data terdekat. Pengambilan data terdekat dapat diambil di 10, 20, atau 30 menit terdekat. Jika dalam kondisi tersebut tidak terdapat data terdekat pada jam yang sama, maka diambil data dari jam sebelum atau jam sesudahnya yang terdekat. Jika dilihat dari data tanggal 1 januari sampai 21 November 2019 terdapat 7723 data. Jika data tidak kosong jumlah data yang seharusnya yaitu 8016 data. Terdapat 334 data kosong dari total 8016 data. Setiap data kosong diambil dari jam terdekat hingga semua data kosong terisi.

Data yang sudah didapatkan kemudian dilakukan pemotongan untuk mengamati suatu daerah tertentu. Penelitian ini memotong citra himawari pada wilayah Kalimantan Selatan dan sekitarnya. Pengambilan wilayah Kalimantan Selatan dikarenakan wilayah kalimantan berada disekitar wilayah katulistiwa.

Berdasarkan teori terbentuknya awan, maka awan sering muncul di daerah katulistiwa. Kemunculan awan di daerah katulistiwa dikarenakan daerah katulistiwa merupakan titik belok angin sebagai faktor terbentuknya awan. Sebagai contoh hasil pemotongan citra himawari ditunjukkan pada Gambar 4.2. Gambar 4.2, terdapat beberapa kumpulan awan cb yang tumbuh disekitar wilayah Kalimantan Selatan.



Gambar 4.2 Contoh Hasil Pemotongan Citra pada Wilayah Kalimantan Selatan, dan Sekitarnya

Proses selanjutnya yaitu mengambil lapisan merah pada citra RGB. Pengambilan citra merah dikarenakan kemunculan awan cb ditandai dengan meningkatnya intensitas warna merah pada citra. Pengambilan lapisan merah dilakukan untuk setiap data setelah dilakukan pemotongan citra. Contoh pengambilan lapisan merah dari Gambar 4.2, ditunjukkan pada Gambar 4.3. Hasil pengambilan lapisan merah menunjukkan semakin tinggi intensitas warna merah maka semakin putih warna yang ditampilkan. Semakin kecil intensitas merah pada citra RGB maka semakin hitam warna yang ditampilkan pada citra merah.



Gambar 4.3 Contoh Hasil Pengambilan Citra Lapisan Merah

Hasil pengambilan citra merah dilanjutkan dengan pembersihan *noise* menggunakan metode median filter. Pembersihan *noise* menggunakan metode median filter dilakukan dengan mengambil nilai tengah dari sejumlah piksel ketetanggaannya. Sebagai contoh penghilangan *noise* pada citra dengan ukuran 10×10 . Misalkan nilai piksel pada citra memiliki nilai seperti pada Tabel 4.1.

Tabel 4.1 Contoh Piksel Citra Ukuran 10 × 10

0	67	67	72	52	52	52	52	10	52
0	67	67	72	52	52	52	52	52	72
0	0	0	0	72	72	72	67	67	0
156	140	140	156	140	140	0	0	0	0
205	197	197	156	140	140	140	140	0	0
255	205	197	156	140	140	140	0	0	0
255	205	197	156	156	140	0	0	0	67
197	140	0	140	0	0	67	72	72	52
140	0	72	67	67	67	72	52	52	10
0	72	52	52	52	72	10	10	10	52

Piksel yang ditunjukkan pada tabel 4.1 diperbaiki menggunakan median filter dengan filter 3×3 . Misalkan perbaikan nilai piksel diberikan untuk memperbaiki piksel pada baris 2 kolom ke 2, maka nilai ketetagganya diambil seberti pada Tabel 4.2. Selanjutnya diurutkan berdasarkan nilai pikselnya dari kecil ke besar. Hasil nilai piksel yang sudah diurutkan diambil nilai tengahnya untuk menggantikan nilai piksel sebelumnya. Hasil median filter pada Tabel 4.2 ditunjukkan pada Tabel 4.3.

Tabel 4.2 Piksel Ketetanggan dari Nilai Tabel4.1 Baris 2, dan Kolom 2

0	67	67	0	67	67	0	0	0
---	----	----	---	----	----	---	---	---

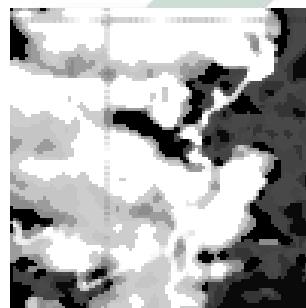
Tabel 4.3 Nilai Piksel Setelah diurutkan

0	0	0	0	0	67	67	67	67
---	---	---	---	---	----	----	----	----

Nilai tengah yang diperoleh yaitu 0, maka nilai 0 menggantikan nilai piksel 67 pada baris ke 2 kolom ke 2. Proses perbaikan piksel tersebut dilakukan untuk semua piksel. Setelah diberikan perbaikan piksel maka didapatkan nilai piksel baru seperti pada Tabel 4.4. Setelah diberikan penghilangan *noise* menggunakan *noise* didapatkan citra seperti pada Gambar 4.4.

Tabel 4.4 Nilai Piksel Citra Setelah Diberikan Median Filter

0	0	67	52	52	52	52	52	52	0
0	0	67	67	52	52	52	52	52	10
0	67	67	72	72	72	52	52	52	0
0	140	140	140	140	140	72	67	0	0
156	197	156	156	140	140	140	0	0	0
205	205	197	156	140	140	140	0	0	0
197	197	156	156	140	140	67	0	0	0
140	140	140	72	67	67	67	52	52	10
0	72	67	52	67	67	67	52	52	10
0	0	52	52	52	52	10	10	10	0

**Gambar 4.4 Contoh Hasil Penghilangan Noise Menggunakan Median Filter**

Setelah semua citra diperbaiki menggunakan median filter citra dirubah menjadi satu vektor kolom sebagai masukan pada proses prediksi. Jumlah piksel yang dimiliki data yaitu 50×50 piksel, sehingga didapatkan satu data mempunyai

panjang vektor 2500×1 . Setelah semua data citra diubah menjadi vektor, selanjutnya data dikumpulkan menjadi dalam satu matriks U . Data pertama menempati kolom pertama pada matriks U . Dilanjutkan data kedua menempati kolom kedua begitu seterusnya hingga semua data terkumpul menjadi satu Matriks U . Jumlah data yang dimiliki yaitu 8014 data, sehingga setelah dikumpulkan menjadi satu diperoleh matriks dengan ukuran 2500×8014 . Banyak baris pada matriks U menunjukkan banyaknya piksel dalam satu data, sedangkan banyaknya kolom menunjukkan banyaknya data yang digunakan. Kumpulan data pada matriks U disajikan pada Tabel 4.5. Setelah semua nilai matriks U didapatkan selanjutnya nilai piksel dapat menjadi nilai masukan dalam proses prediksi.

Tabel 4.5 Data Pada Matriks U

4.2. Prediksi Menggunakan *Deep Echo State Network* (DeepESN)

Prediksi pertumbuhan awan dilakukan untuk melihat kondisi awan satu jam kedepan. Proses prediksi menggunakan metode DeepESN. Prediksi menggunakan DeepESN diawali dengan proses pelatihan DeepESN. Pelatihan DeepESN ditujukan untuk mendapatkan bobot keluaran (w_{out}). Pada proses pelatihan, panjang data yang digunakan bermacam macam. Uji coba pada proses pelatihan diberikan untuk melihat pengenalan pola yang optimal menggunakan DeepESN. Sebelum melakukan pelatihan DeepESN diperlukan beberapa inisialisasi awal. Inisialisasi yang pertama yaitu menentukan jumlah masukan atau N_u . Pada penelitian ini memiliki ukuran masukan sesuai dengan banyaknya kolom pada matriks U . Didapatkan ukuran masukan (N_u) = 2500. Selanjutnya yaitu menentukan banyaknya *node* pada tiap *reservoir*, dan banyaknya lapisan *reservoir*. Penelitian ini menentukan jumlah *node*, dan banyaknya lapisan *reservoir* dengan menggunakan beberapa percobaan. Percobaan banyaknya *node* (N_r) menggunakan 3 percobaan yaitu 100, 120, dan 150 *node*. Untuk percobaan banyaknya lapisan *reservoir* (N_l), dilakukan menggunakan 3 percobaan yaitu, 100, 150, dan 200 lapisan. Selain percobaan pada jumlah *node*, dan jumlah lapisan, uji coba juga dilakukan untuk menentukan nilai *learning rate* (α). Nilai α diberikan 3 uji coba yaitu 0.5, 0.8, dan 0.9. Uji coba dilakukan untuk melihat *learning rate* yang paling baik. Inisialisasi selanjutnya yaitu ukuran keluaran (N_y). Karena penelitian ini ditujukan untuk melihat kondisi citra dengan ukuran yang sama, maka ukuran keluaran $N_y = N_u = 2500$ *node*.

Proses pelatihan DeepESN dilakukan dengan memasukkan data pelatihan satu persatu menuju lapisan *readout*. Data yang dimasukkan pada DeepESN diambil satu persatu dari matriks U . Data pertama yang dimasukkan yaitu data

pada semua baris di kolom pertama. Dilanjutkan untuk masukan kedua yaitu semua baris pada kolom ke-2 dari matriks U . Jumlah data yang digunakan pada proses pelatihan yaitu mulai dari 1 Januari hingga 31 Agustus 2019 yang diambil setiap satu jam sekali. Sehingga didapatkan data sebanyak 5832 data. Sedangkan untuk data pengujian berlangsung mulai tanggal 1 September hingga 30 November 2019 atau sebanyak 2184 data. Sebelum data pelatihan dimasukkan pada proses pelatihan DeepESN terlebih dahulu dilakukan inisialisasi bobot-bobot yang ada dalam DeepESN.

Jenis bobot yang terdapat pada DeepESN terdiri dari 5 macam bobot. Nilai bobot yang pertama yaitu w_{in} . w_{in} menghubungkan setiap *node* yang ada pada lapisan masukan menuju *node* yang ada pada *reservoir* lapisan pertama. Pada percobaan yang pertama menggunakan $\alpha = 0.5$, $N_r = 100$, dan $N_l = 100$. Sedangkan jumlah masukan pada penelitian ini yaitu 2500, sehingga diperoleh ukuran bobot w_{in} yaitu $N_r \times N_u = 100 \times 2500$. Nilai pada matriks w_{in} dibangkitkan secara acak dengan interval -1 sampai 1. Setelah nilai w_{in} dibangkitkan secara acak diperoleh nilai pada matriks w_{in} seperti pada Tabel 4.6.

Tabel 4.6 Nilai Matriks Bobot Masukan(w_{in})

-2.66×10^{-5}	-4.07×10^{-5}	-8.19×10^{-5}	...	-4.54×10^{-5}
9.34×10^{-5}	8.22×10^{-5}	-3.94×10^{-5}	...	7.42×10^{-5}
-6.47×10^{-5}	9.65×10^{-5}	-7.84×10^{-6}	...	-7.14×10^{-6}
5.40×10^{-5}	6.87×10^{-5}	-3.57×10^{-5}	...	-2.27×10^{-5}
3.82×10^{-6}	-3.01×10^{-5}	-4.66×10^{-5}	...	-9.09×10^{-5}
2.72×10^{-5}	-2.94×10^{-5}	-8.80×10^{-6}	...	-6.14×10^{-5}
-9.33×10^{-5}	-9.10×10^{-5}	6.28×10^{-5}	...	5.47×10^{-5}
-2.10×10^{-6}	7.68×10^{-5}	-1.57×10^{-5}	...	9.35×10^{-5}
⋮	⋮	⋮	⋮	⋮
8.34×10^{-5}	-3.27×10^{-5}	6.08×10^{-5}	...	-6.24×10^{-5}

Setelah mendapatkan nilai bobot masukan, dilanjutkan inisialisasi bobot untuk menghubungkan antar *reservoir* yang bersebelahan. Bobot yang

menghubungkan antar *reservoir* yaitu w_l . Banyaknya w_l yang dibutuhkan sesuai dengan banyaknya keterhubungan *reservoir* yang bersebelahan. Pada percobaan pertama banyak *reservoir* yang digunakan sebanyak 100 lapisan. Ketika lapisan *reservoir* yang digunakan sebanyak 100, maka keterkaitan yang diperlukan yaitu 99. Banyaknya w_l juga dapat dihitung dengan cara melihat banyaknya *reservoir* (N_l) – 1. Nilai pada bobot w_l menghubungkan setiap *node* pada *reservoir* satu dengan setiap *node* pada *reservoir* berikutnya. Karena banyak *node* pada percobaan yang pertama yaitu 100, dan banyaknya lapisan(N_l) yaitu 100. Diperoleh ukuran w_l yaitu $100 \times 100 \times 99$. Ukuran 100×100 menunjukkan ukuran keterkaitan untuk setiap *node*, sedangkan 99 menunjukkan banyaknya keterkaitan antar *reservoir*. Pembangkitan bobot w_l sama seperti pada bobot masukan yaitu dengan membangkitkan secara acak. Berikut merupakan hasil pembangkitan bobot w_l lapisan pertama menuju lapisan kedua yang ditunjukkan pada Tabel 4.7. Berikutnya yaitu pembangkitan bobot w_l yang menghubungkan lapisan ke-dua menuju lapisan ke-tiga ditunjukkan pada Tabel 4.8. Pembangkitan bobot w_l dilakukan hingga w_l ke-99 yang menghubungkan antara lapisan 99 menuju lapisan terakhir yaitu 100. Pembangkitan bobot w_l ke-99 ditunjukkan pada Tabel 4.9. Masing-masing tabel merupakan sampel dari w_l pada lapisan $l = 1$, $l = 2$, dan $l = 99$.

Jenis Bobot yang ketiga yaitu w_x . Bobot w_x menghubungkan *node* yang ada dalam satu *reservoir* menuju *node* lainnya, termasuk terkait dengan dirinya sendiri. Jika jumlah *node* yang digunakan sebanyak 100 *node*, maka diperoleh jumlah keterkaitan yaitu 100×100 . Bobot w_x dibangkitkan secara acak seperti pada bobot lainnya. Berikut merupakan hasil pembangkitan bobot w_x , dengan 100 *node* pada lapisan *reservoir* ke-1 ditunjukkan pada Tabel 4.10, dan lapisan

Tabel 4.7 Nilai Matriks $w_{l=1}$

-3.49×10^{-5}	8.8×10^{-5}	-5.00×10^{-6}	...	8.66×10^{-5}
-3.38×10^{-6}	1.07×10^{-5}	4.44×10^{-5}	...	-6.92×10^{-5}
-8.71×10^{-5}	-4.96×10^{-5}	4.84×10^{-5}	...	5.35×10^{-5}
-2.73×10^{-6}	1.53×10^{-5}	5.18×10^{-5}	...	-2.27×10^{-5}
6.70×10^{-5}	4.43×10^{-5}	7.41×10^{-5}	...	-7.48×10^{-5}
8.02×10^{-5}	-4.18×10^{-5}	-3.23×10^{-5}	...	5.63×10^{-5}
8.97×10^{-7}	9.38×10^{-5}	3.58×10^{-7}	...	3.76×10^{-5}
-1.53×10^{-5}	2.15×10^{-5}	6.32×10^{-6}	...	-1.07×10^{-5}
⋮	⋮	⋮	⋮	⋮
-2.76×10^{-5}	7.64×10^{-5}	-5.63×10^{-5}	...	-1.51×10^{-6}

Tabel 4.8 Nilai Matriks $w_{l=2}$

2.67×10^{-6}	-8.09×10^{-5}	2.25×10^{-5}	...	2.34×10^{-5}
4.07×10^{-6}	-2.86×10^{-5}	-5.73×10^{-5}	...	3.29×10^{-6}
4.09×10^{-5}	-8.14×10^{-5}	3.63×10^{-5}	...	8.10×10^{-5}
4.44×10^{-6}	8.71×10^{-5}	-6.69×10^{-5}	...	4.20×10^{-5}
-9.89×10^{-5}	-3.54×10^{-5}	-6.96×10^{-5}	...	9.20×10^{-5}
9.00×10^{-5}	6.71×10^{-5}	-7.03×10^{-5}	...	-2.18×10^{-5}
7.20×10^{-5}	6.27×10^{-5}	7.11×10^{-5}	...	6.38×10^{-5}
-1.54×10^{-5}	2.41×10^{-6}	6.97×10^{-5}	...	7.86×10^{-5}
⋮	⋮	⋮	⋮	⋮
-3.25×10^{-5}	2.16×10^{-5}	5.30×10^{-5}	...	-5.58×10^{-5}

reservoir ke-2 ditunjukkan pada Tabel 4.11. Bobot w_x dibangkitkan hingga lapisan yang terakhir yaitu pada lapisan $l = 100$. Nilai w_x pada lapisan terakhir ditunjukkan pada Tabel 4.12.

Nilai bobot selanjutnya yaitu bobot bias (w_θ) pada masing-masing lapisan *reservoir*. Setiap bias terhubung dengan masing-masing *node* yang ada dalam satu *reservoir*. Bobot bias menghubungkan nilai bias pada setiap *node* yang ada pada *reservoir*. Pada percobaan yang pertama menggunakan *node* yang berjumlah 100, dengan 100 lapisan maka diperoleh w_θ dengan ukuran 100×100 . Berikut merupakan hasil nilai bobot w_θ yang dibangkitkan secara acak yang disajikan pada Tabel 4.13. Nilai w_θ yang disajikan pada Tabel 4.13 memiliki jumlah baris 100

Tabel 4.9 Nilai Matriks $w_{l=99}$

9.58×10^{-5}	-6.63×10^{-5}	8.49×10^{-5}	...	-6.68×10^{-5}
1.37×10^{-5}	-6.62×10^{-6}	-2.76×10^{-5}	...	-6.02×10^{-5}
7.50×10^{-5}	-9.55×10^{-5}	2.79×10^{-5}	...	4.96×10^{-6}
-2.66×10^{-5}	-1.91×10^{-5}	-3.14×10^{-5}	...	4.90×10^{-6}
-1.90×10^{-5}	-1.29×10^{-5}	4.37×10^{-5}	...	-9.78×10^{-5}
4.38×10^{-5}	-6.58×10^{-5}	-8.29×10^{-6}	...	-9.04×10^{-5}
5.27×10^{-5}	-4.53×10^{-5}	-3.72×10^{-5}	...	4.76×10^{-5}
6.85×10^{-5}	4.26×10^{-5}	8.88×10^{-5}	...	-6.85×10^{-5}
⋮	⋮	⋮	⋮	⋮
6.22×10^{-5}	2.27×10^{-5}	-5.02×10^{-5}	...	6.47×10^{-5}

Tabel 4.10 Nilai Matriks w_x Pada Lapisan 1

-0.730	0.226	-0.046	0.197	...	-0.128
0.251	-0.964	-0.153	0.008	...	-0.112
-0.268	0.095	-0.458	0.159	...	0.217
-0.086	-0.208	-0.097	-0.994	...	0.188
0.237	0.263	-0.159	0.066	...	0.134
-0.269	-0.233	-0.158	0.064	...	0.101
0.129	-0.169	0.0310	-0.092	...	-0.171
0.061	0.141	-0.216	-0.225	...	0.083
⋮	⋮	⋮	⋮	⋮	⋮
0.065	-0.049	0.252	0.123	...	-0.495

yang menunjukkan jumlah keterkaitan pada semua *node* dalam satu *reservoir*. Sedangkan Jumlah kolom menunjukkan jumlah lapisan yang terhubung dengan bias.

Bobot yang terakhir yaitu bobot keluaran w_{out} . Pada tahap pelatihan nilai w_{out} didapatkan dengan melihat nilai keluaran pada setiap *reservoir* ($x(t)$), dan target yang diinginkan. Awalnya nilai $x(t)$ pada waktu $t = 0$ diinisialisasikan dengan 0. Selanjutnya, nilai $x(t)$ didapatkan dari perhitungan yang ada pada setiap reservoir. Pada data pertama ($t = 1$) yang ada pada matriks U dihitung untuk mendapatkan nilai $x(t)$ *reservoir* pertama. Nilai $x(t)$ pada lapisan pertama ($l = 1$)

Tabel 4.11 Nilai Matriks w_x Pada Lapisan 2

-0.723	-0.126	-0.144	-0.121	...	0.070
-0.133	-0.796	-0.009	0.241	...	-0.125
-0.185	-0.180	-0.889	-0.140	...	-0.200
0.228	-0.250	-0.033	-0.711	...	0.275
0.236	0.167	-0.212	0.057	...	-0.144
0.269	0.044	-0.053	0.150	...	0.054
-0.030	-0.272	-0.210	0.053	...	-0.009
0.118	-0.186	-0.045	0.257	...	0.150
⋮	⋮	⋮	⋮	⋮	⋮
0.248	-0.001	-0.263	-0.001	...	-0.508

Tabel 4.12 Nilai Matriks w_x Pada Lapisan 100

-0.662	-0.090	0.196	-0.246	...	0.243
-0.169	-0.619	-0.143	0.143	...	-0.228
0.093	0.185	-0.877	-0.084	...	0.136
-0.015	0.015	-0.050	-0.853	...	0.216
0.089	0.072	0.244	0.039	...	0.210
-0.170	-0.199	-0.015	-0.091	...	0.078
0.243	0.140	0.246	-0.139	...	0.158
0.207	0.147	0.224	0.156	...	-0.097
⋮	⋮	⋮	⋮	⋮	⋮
-0.162	0.174	-0.074	-0.245	...	-0.652

$(x_t^{(1)})$ didapatkan seperti berikut:

$$x^{(1)}(t) = (1 - \alpha^{(1)})x^{(1)}(t - 1) + \alpha^{(1)}f(w_{in}^{(1)}i^{(1)}(t) + w_\theta^{(1)}\theta^{(1)} + w_x^{(1)}x^{(1)}(t - 1))$$

Karena $l = 1$, maka diperoleh $i^{(l)} = U(t)$. Sedangkan disaat $t = 1$ maka data yang digunakan yaitu semua baris pada kolom pertama dari matriks U ($U(1)$). Nilai $U(1)$

dapat dituliskan sebagai berikut:

Selanjutnya dengan memperhatikan kondisi awal $x(t = 0) = 0$,

berpengaruh pada nilai $(1 - \alpha^{(1)})x^{(1)}(t - 1)$ meghasilkan nilai

$$(1 - \alpha^{(1)})x^{(1)}(t - 1) = 0$$

Kondisi $x(t = 0) = 0$ berpengaruh juga pada proses perhitungan $w_x^{(l)} x^{(l)}(t - 1)$, sehingga didapatkan nilai sebagai berikut:

$$w_x^{(l)} x^{(l)}(t-1) = 0$$

Sedangkan pada nilai $w_{in}U(1)$ diperoleh dengan memerhatikan nilai w_{in} pada Tabel 4.6, dan Nilai $U(1)$.

$$w_{in} U(1) = \begin{bmatrix} -2.66 \times 10^{-5} & -4.07 \times 10^{-5} & \dots & -4.54 \times 10^{-5} \\ 9.34 \times 10^{-5} & 8.22 \times 10^{-5} & \dots & 7.42 \times 10^{-5} \\ -6.47 \times 10^{-5} & 9.65 \times 10^{-5} & \dots & -7.14 \times 10^{-6} \\ 5.40 \times 10^{-5} & 6.87 \times 10^{-5} & \dots & -2.27 \times 10^{-5} \\ 3.82 \times 10^{-6} & -3.01 \times 10^{-5} & \dots & -9.09 \times 10^{-5} \\ 2.72 \times 10^{-5} & -2.94 \times 10^{-5} & \dots & -6.14 \times 10^{-5} \\ -9.33 \times 10^{-5} & -9.10 \times 10^{-5} & \dots & 5.47 \times 10^{-5} \\ -2.10 \times 10^{-6} & 7.68 \times 10^{-5} & \dots & 9.35 \times 10^{-5} \\ \vdots & \vdots & \vdots & \ddots \\ 8.34 \times 10^{-5} & -3.27 \times 10^{-5} & \dots & -6.24 \times 10^{-5} \end{bmatrix} \times \begin{bmatrix} 0 \\ 67 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 140 \\ 140 \\ \vdots \\ 0 \end{bmatrix}$$

Selanjutnya nilai $w_\theta^{(1)}\theta^{(1)}$ dengan nilai $\theta = 1$ diperoleh nilai $w_\theta^{(1)}\theta^{(1)} = w_\theta^{(1)}$
 Sesuai pada Tabel 4.13. Melihat dari beberapa hasil perhitungan diatas didapatkan

nilai $x(1)^{(1)}$ sebagai berikut:

$$x^{(1)}(1) = 0 + 0.9 \tanh \left(\begin{array}{c|c|c} -0, 201 & -4.82 \times 10^{-5} & -0, 198 \\ 0, 945 & -8.83 \times 10^{-5} & 0, 737 \\ 0, 022 & 3.96 \times 10^{-5} & 0, 022 \\ 0, 249 & 7.95 \times 10^{-5} & 0, 244 \\ 0, 218 & -8.28 \times 10^{-5} & 0, 215 \\ 0, 239 & 5.97 \times 10^{-5} & 0, 234 \\ -0, 218 & -5.55 \times 10^{-5} & -0, 21 \\ 0, 740 & 3.15 \times 10^{-5} & 0, 629 \\ \vdots & \vdots & \vdots \\ -0, 058 & 8.69 \times 10^{-5} & -0, 058 \end{array} \right) = 0$$

Proses selanjutnya yaitu membawa nilai $x(1)^{(1)}$ sebagai nilai masukan lapisan ke-2 ($l = 2$), untuk memperoleh nilai $x(1)^{(2)}$. Seperti pada perhitungan nilai $x(1)^{(1)}$, nilai $x(1)^{(2)}$ diperoleh menggunakan yang sama. Perbedaannya yaitu nilai $i^{(l)} = x(1)^{(1)}$. Nilai dari *reservoir* pertama menjadi masukan pada *reservoir* kedua. Seperti pada perhitungan *reservoir* pertama, kondisi awal $x(t = 0) = 0$. Sehingga didapatkan nilai dari $w_x^{(2)}x^{(2)}(0) = 0$. Berikut proses perhitungan untuk $x^{(2)}(1)$.

$$x^{(2)}(1) = 0 + 0.9 \tanh \left(\begin{bmatrix} 0,0002 \\ -8,08 \times 10^{-5} \\ -2,28 \times 10^{-5} \\ 0,0001 \\ \vdots \\ 0,0001 \end{bmatrix} + \begin{bmatrix} -9.42 \times 10^{-5} \\ 4.29 \times 10^{-5} \\ -7.94 \times 10^{-5} \\ -2.16 \times 10^{-5} \\ \vdots \\ -1.28 \times 10^{-5} \end{bmatrix} + 0 \right) = \begin{bmatrix} 0,0002 \\ -8,08 \times 10^{-5} \\ -2,28 \times 10^{-5} \\ 0,0001 \\ \vdots \\ 0,0001 \end{bmatrix}$$

Perhitungan yang sama dilakukan hingga lapisan terakhir yaitu lapisan ke-100. Untuk memperoleh $x^{(100)}(1)$. Nilai $x^{(100)}(1)$ menjadikan $x^{(99)}(1)$ sebagai nilai masukan pada lapisan *reservoir* ke-100. Sehingga nilai $x^{(100)}(1)$ didapatkan seperti berikut:

$$x^{(100)}(1) = 0 + 0.9 \tanh \left(\begin{pmatrix} -9.42 \times 10^{-5} \\ 4.29 \times 10^{-5} \\ -7.94 \times 10^{-5} \\ -2.16 \times 10^{-5} \\ -4.39 \times 10^{-5} \\ -4.96 \times 10^{-5} \\ \vdots \\ -1.28 \times 10^{-5} \end{pmatrix} + 0 \right) = \begin{pmatrix} -9.42 \times 10^{-5} \\ 4.29 \times 10^{-5} \\ -7.94 \times 10^{-5} \\ -2.15 \times 10^{-5} \\ -4.39 \times 10^{-5} \\ -4.95 \times 10^{-5} \\ \vdots \\ -1.28 \times 10^{-5} \end{pmatrix}$$

Selanjutnya proses perhitungan dilanjutkan pada saat $t = 2$. Misalkan data saat

Tabel 4.13 Nilai Matriks w_θ

$w_\theta^{(1)}$	$w_\theta^{(2)}$	$w_\theta^{(3)}$...	$w_\theta^{(100)}$
-4.82×10^{-5}	-2.12×10^{-5}	-6.01×10^{-5}	...	-9.42×10^{-5}
-8.83×10^{-5}	-5.22×10^{-5}	4.96×10^{-5}	...	4.29×10^{-5}
3.96×10^{-5}	4.89×10^{-5}	8.65×10^{-5}	...	-7.94×10^{-5}
7.95×10^{-5}	7.34×10^{-5}	-7.79×10^{-5}	...	-2.16×10^{-5}
-8.28×10^{-5}	4.40×10^{-5}	-5.52×10^{-5}	...	-4.39×10^{-5}
5.97×10^{-5}	-6.49×10^{-5}	-9.68×10^{-5}	...	-4.96×10^{-5}
-5.55×10^{-5}	-8.94×10^{-5}	7.22×10^{-5}	...	-6.01×10^{-6}
3.15×10^{-5}	8.22×10^{-5}	-9.20×10^{-5}	...	6.05×10^{-6}
⋮	⋮	⋮	⋮	⋮
8.69×10^{-5}	9.93×10^{-5}	-5.35×10^{-5}	...	-1.28×10^{-5}

$t = 2$ sebagai berikut:

Seperti pada proses perhitungan data $U(1)$, data $U(2)$ dipetakan menuju masing masing *reservoir* dari lapisan satu sampai lapisan terakhir. Karena $x(t-1)$ tidak lagi sama dengan 0 maka nilai dari $(1-\alpha^{(1)})x^{(1)}(t-1)$ tidak sama dengan nol. Sehingga perlu diperhatikan proses perhitungannya. Menggunakan nilai $x(1)^{(1)}$ pada proses sebelumnya, didapatkan hasil sebagai berikut:

$$(1 - 0.9)x^{(1)}(1) = 0.1 \times \begin{bmatrix} -0,198 \\ 0,737 \\ 0,022 \\ 0,244 \\ 0,215 \\ 0,234 \\ \vdots \\ -0,058 \end{bmatrix} = \begin{bmatrix} -0,019 \\ 0,073 \\ 0,002 \\ 0,024 \\ 0,021 \\ 0,023 \\ \vdots \\ -0,005 \end{bmatrix}$$

Perlu diperhatikan juga untuk menghitung $w_x^{(l)}x^{(l)}(t - 1)$ tidak lagi sama dengan nol. Menggunakan data masukan kedua didapatkan hasil dari perhitungan tersebut yaitu

$$w_x^{(1)}x^{(1)}(1) = \begin{bmatrix} -0.730 & 0.226 & \cdots & -0.128 \\ 0.251 & -0.964 & \cdots & -0.112 \\ -0.268 & 0.095 & \cdots & 0.217 \\ -0.086 & -0.208 & \cdots & 0.188 \\ 0.237 & 0.263 & \cdots & 0.134 \\ -0.269 & -0.233 & \cdots & 0.101 \\ \vdots & \vdots & \ddots & \vdots \\ 0.065 & -0.049 & \cdots & -0.495 \end{bmatrix} \times \begin{bmatrix} -0,198 \\ 0,737 \\ 0,022 \\ 0,244 \\ 0,215 \\ 0,234 \\ \vdots \\ -0,058 \end{bmatrix} = \begin{bmatrix} 0.123 \\ -1.600 \\ -0.264 \\ -1.066 \\ 0.452 \\ -0.839 \\ \vdots \\ -0.001 \end{bmatrix}$$

Selanjutnya untuk nilai $w_{in} \times U(2)$ diberoleh sebagai berikut:

$$w_{in}U(2) = \begin{bmatrix} -2.66 \times 10^{-5} & -4.07 \times 10^{-5} & \cdots & -4.54 \times 10^{-5} \\ 9.34 \times 10^{-5} & 8.22 \times 10^{-5} & \cdots & 7.42 \times 10^{-5} \\ -6.47 \times 10^{-5} & 9.65 \times 10^{-5} & \cdots & -7.14 \times 10^{-6} \\ 5.40 \times 10^{-5} & 6.87 \times 10^{-5} & \cdots & -2.27 \times 10^{-5} \\ 3.82 \times 10^{-6} & -3.01 \times 10^{-5} & \cdots & -9.09 \times 10^{-5} \\ 2.72 \times 10^{-5} & -2.94 \times 10^{-5} & \cdots & -6.14 \times 10^{-5} \\ -9.33 \times 10^{-5} & -9.10 \times 10^{-5} & \cdots & 5.47 \times 10^{-5} \\ -2.10 \times 10^{-6} & 7.68 \times 10^{-5} & \cdots & 9.35 \times 10^{-5} \\ \vdots & \vdots & \ddots & \vdots \\ 8.34 \times 10^{-5} & -3.27 \times 10^{-5} & \cdots & -6.24 \times 10^{-5} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 140 \\ 140 \\ 140 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} -0.103 \\ 0.927 \\ -0.497 \\ 0.778 \\ -0.300 \\ 0.036 \\ 0.119 \\ 0.822 \\ \vdots \\ 0.5251 \end{bmatrix}$$

Setelah beberapa bagian perhitungan selesai, nilai-nilai yang didapat dimasukkan ke dalam fungsi untuk mencari nilai $x(2)^{(1)}$ sebagai berikut:

$$x^{(1)}(2) = \begin{bmatrix} -0.019 \\ 0.073 \\ 0.002 \\ 0.024 \\ 0.021 \\ 0.023 \\ -0.021 \\ 0.062 \\ \vdots \\ -0.005 \end{bmatrix} + 0.9 \tanh \begin{bmatrix} -0.103 \\ 0.927 \\ -0.497 \\ 0.778 \\ -0.300 \\ 0.036 \\ 0.119 \\ 0.822 \\ \vdots \\ 0.5251 \end{bmatrix} + \begin{bmatrix} -9.42 \times 10^{-5} \\ 4.29 \times 10^{-5} \\ -7.94 \times 10^{-5} \\ -2.16 \times 10^{-5} \\ -4.39 \times 10^{-5} \\ -4.96 \times 10^{-5} \\ -6.01 \times 10^{-6} \\ 6.05 \times 10^{-6} \\ \vdots \\ -1.28 \times 10^{-5} \end{bmatrix} = \begin{bmatrix} 0.123 \\ -1.600 \\ -0.264 \\ -1.066 \\ 0.452 \\ -0.839 \\ -0.245 \\ -1.167 \\ \vdots \\ -0.001 \end{bmatrix} = \begin{bmatrix} -0.078 \\ -0.218 \\ -0.630 \\ -0.158 \\ 0.258 \\ -0.547 \\ -0.232 \\ -0.016 \\ \vdots \\ 0.450 \end{bmatrix}$$

Perhitungan $x(t)^{(l)}$ berlangsung secara terus menerus hingga untuk setiap data pelatihan sudah mendapatkan nilai $x(t)^{(l)}$. Pada iterasi ke-2 sampai terakhir nilai $x(t - 1)^{(l)}$ tidak lagi bernilai 0. nilai $x(t)^{(l)}$ pada data sebelumnya menjadi masukan nilai $x(t)^{(l)}$ pada data selanjutnya. Sehingga didapatkan antara $x(t - 1)^{(l)}$, $x(t)^{(l)}$ saling berpengaruh. Pengaruhnya nilai $x(t)^{(l)}$ menjadi salah satu nilai yang dapat mempelajari pola untuk melakukan prediksi. Semakin banyak data training yang digunakan maka semakin banyak $x(t)^{(l)}$ mempelajari pola dari data pelatihan. Selanjutnya untuk mendapatkan bobot keluaran (w_{out}) yang optimal disimpan nilai $x(t)^{(l)}$ pada setiap data masukan. Pada penelitian ini data pelatihan yang digunakan sebanyak 5832 data. Setiap data $x(t)^{(l)}$ disimpan dalam matiks X hingga matriks X memiliki ukuran 5832×10000 .

Nilai X digunakan untuk pencarian bobot yang optimal. Untuk

mendapatkan nilai bobot yang optimal terlebih dahulu dihitung nilai $(X'X)^{-1}$. Didapatkan nilai X dengan ukuran 10000×10000 . Nilai yang dihasilkan dikalikan dengan nilai target data. Sehingga diperoleh nilai w_{out} yang optimal dengan ukuran 2500×10000 . Misalkan dari proses pelatihan didapatkan w_{out} seperti berikut:

$$w_{out} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 2,664 & -9,353987861 & -31,174 & 6,171 & 9,371 & \dots & -5.360 \\ -1,968 & -3,611 & -25,483 & 8,859 & 10,934 & \dots & -4.286 \\ -5,814 & 10,254 & -18,663 & 15,013 & 6,636 & \dots & -5.4660 \\ -3,886 & -3,538 & -0,5926 & 9,741 & 11,150 & \dots & -4.702 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

Nilai w_{out} yang optimal dijadikan sebagai w_{out} pada proses *testing*. Penelitian ini menggunakan beberapa percobaan pelatihan dan *testing* untuk mencari struktur DeepESN yang paling baik. Percobaan pelatihan diberikan perlakuan berbeda pada beberapa parameter. Parameter yang diberikan perlakuan berbeda diantaranya yaitu nilai *learning rate* (α), jumlah *node* dalam *reservoir* (N_r), dan jumlah lapisan *reservoir* (N_l). Nilai α yang diberikan yaitu 0.5, 0.8, dan 0.9. Sedangkan untuk perlakuan berbeda pada nilai N_r diberikan jumlah *node* 100, 150, dan 200 *node*. Jumlah lapisan *reservoir* yang diberikan yaitu 100, 120, dan 150 lapisan.

Hasil pengujian menggunakan struktur DeepESN yang paling optimal dilihat dengan menggunakan kemiripan citra asli dengan hasil prediksi. Sebagai nilai tolak ukur kemiripan citra digunakan metode PSNR untuk menghitung

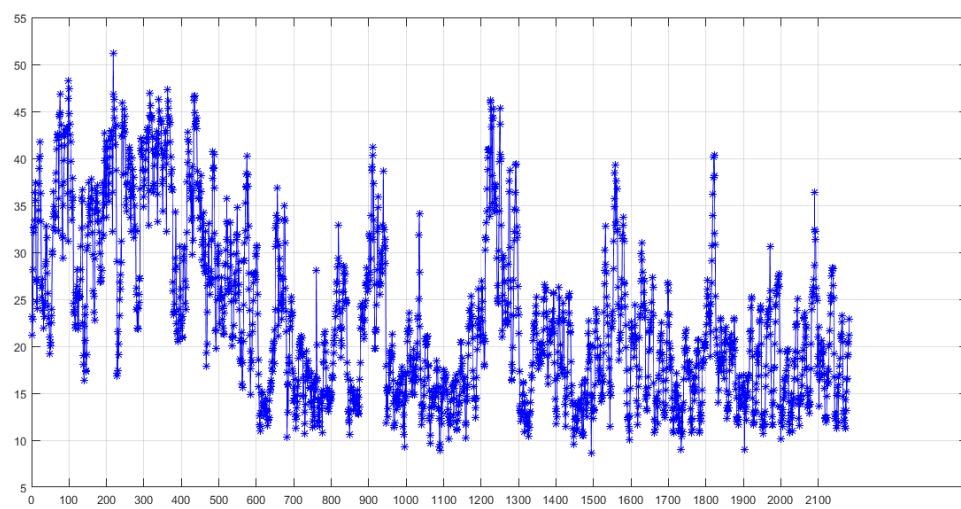
seberapa mirip kedua citra. kemiripan menggunakan PSNR dikeluarkan dengan satuan *decibel* (dB). Berikut merupakan hasil beberapa percobaan dengan menggunakan beberapa perlakuan berbeda yang disajikan pada Tabel 4.14. Nilai PSNR yang disajikan pada Tabel 4.14 merupakan rata rata PSNR dari semua data *testing*.

Tabel 4.14 Hasil Rata Rata PSNR dari Beberapa Percobaan

Percobaan ke-	α	N_r	N_l	Rata Rata Nilai PSNR
1	0.5	100	100	22.9344
2	0.5	100	150	22.9262
3	0.5	100	200	22.9333
4	0.5	120	100	22.8952
5	0.5	120	150	22.994
6	0.5	120	200	22.9051
7	0.5	150	100	22.8271
8	0.5	150	150	22.7795
9	0.5	150	200	22.7993
10	0.8	100	100	22.6873
11	0.8	100	150	22.6425
12	0.8	100	200	22.7526
13	0.8	120	100	22.7256
14	0.8	120	150	22.6127
15	0.8	120	200	22.7045
16	0.8	150	100	22.4701
17	0.8	150	150	22.5632
18	0.8	150	200	22.6936
19	0.9	100	100	22.5939
20	0.9	100	150	22.5627
21	0.9	100	200	22.4424
22	0.9	120	100	22.5893
23	0.9	120	150	22.5351
24	0.9	120	200	22.3839
25	0.9	150	100	22.3389
26	0.9	150	150	22.4719
27	0.9	150	200	22.4119

Melihat dari hasil yang ditunjukkan Tabel 4.14, didapatkan nilai PSNR terbaik berada pada percobaan ke 5. Percobaan terbaik menggunakan nilai

$\alpha = 0.5$, dengan jumlah *node* 120, dan jumlah lapisan *reservoir* sebanyak 150 lapisan. Jika dilihat dari nilai rata rata PSNR terlihat bahwa banyak atau sedikit jumlah lapisan *reservoir* tidak berimbang pada semakin tingginya nilai kesamaan citra. Seperti pada percobaan ke 25, nilai PSNR yang dihasilkan merupakan nilai terkecil sekalipun menggunakan jumlah *node* dan lapisan yang cukup banyak. Dibutuhkan sebuah kombinasi yang tepat untuk mendapatkan hasil yang optimal. Pencarian kombinasi yang tepat dapat diakukan dengan uji coba ketiga parameter seperti pada Tabel 4.14. Hasil dari beberapa kombinasi beberapa nilai parameter didapatkan nilai rata rata PSNR terbaik yaitu 22.994. Nilai rata rata PSNR 22.994 didapatkan dari sebaran nilai PSNR data *testing* pertama hingga terakhir. Hasil uji pada percobaan ke 5 didapatkan nilai PSNR terbaik yaitu 51.2006 dB, dan Nilai terkecil yaitu 8.6763 dB. Sebaran nilai PSNR yang dilakukan pada percobaan ke 5 disajikan pada Tabel 4.15.



Gambar 4.5 Sebaran nilai PSNR Pada Percobaan ke 5

Pada hasil pengujian menggunakan struktur kelima didapatkan hasil yang relatif tinggi dari data pertama hingga sekitar data ke 600. Untuk lebih mudah

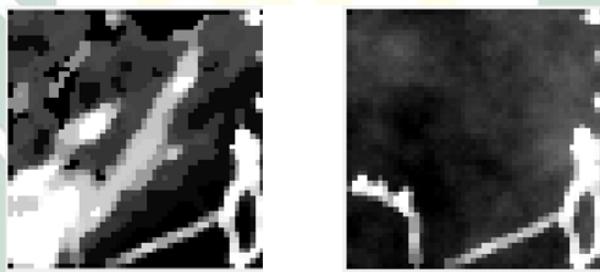
melihat hasil sebaran nilai PSNR ditunjukkan pada Gambar 4.5. Sumbu mendatar menunjukkan data yang dihitung nilai PSNR, sedangkan sumbu vertikal menunjukkan nilai PSNR yang didapatkan. Melihat dari Gambar 4.5 data memiliki kemiripan yang cukup tinggi pada 600 data awal atau citra pada bulan September. Nilai PSNR mengalami penurunan karena terdapat pergantian musim dari musim kemarau menuju musim hujan. Pada saat peralihan musim terjadi perubahan cuaca yang tidak menentu. Perubahan cuaca yang sangat cepat dan tidak menentu membuat sistem yang sudah dibuat sedikit mengalami gangguan.

Tabel 4.15 Sebaran Nilai PSNR pada Percobaan ke 5

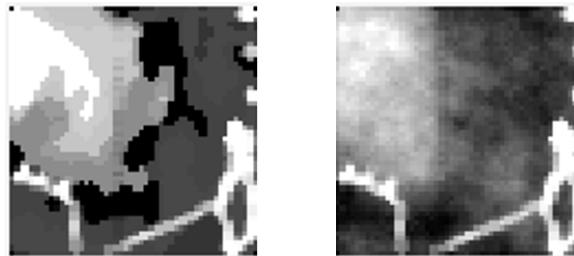
Data ke	Nilai PSNR
1	23.1759
2	21.1488
3	22.9065
4	28.1847
5	32.6639
6	32.0880
7	32.5831
8	33.4677
9	36.3724
10	37.4504
:	:
216	32.1732
217	36.3748
218	51.2005
219	46.8493
220	46.2748
:	:
2181	20.2012
2182	21.1366
2183	22.8814

Pembelajaran pola dengan memperbaiki nilai $x(t)$ setiap kali data baru dimasukkan. Namun, hasil percobaan pengenalan pola dengan perbaikan nilai $x(t)$ tidak cukup untuk meningkatkan nilai PSNR kembali. Diperlukan perbaikan nilai

w_{out} untuk mempelajari pola data yang tidak menentu. Jika dilihat dari hasilnya hasil prediksi citra dengan nilai PSNR paling rendah ditunjukkan pada Gambar 4.6 bagian kanan. Jika dilihat citra yang seharusnya didapatkan oleh hasil prediksi ditunjukkan pada Gambar 4.6 bagian kiri. Citra yang dihasilkan sangat jauh berbeda dengan citra yang seharusnya diprediksi. Kemiripan yang sangat rendah membuat nilai PSNR yang dihasilkan rendah. Hasil prediksi jauh berbeda dikarenakan pengisian data kosong menggunakan data pada jam terdekat. Oleh sebab itu, ketika data kosong diisi dengan data sebelumnya menghasilkan dua citra yang tidak mengalami pergerakan. Sedangkan pola yang dipelajari oleh DeepESN sebuah citra awan mengalami pergerakan setiap jamnya. Mengacu pada kondisi tersebut menyebabkan beberapa data memiliki nilai PSNR yang rendah karena memiliki citra awan yang tidak mengalami perubahan.



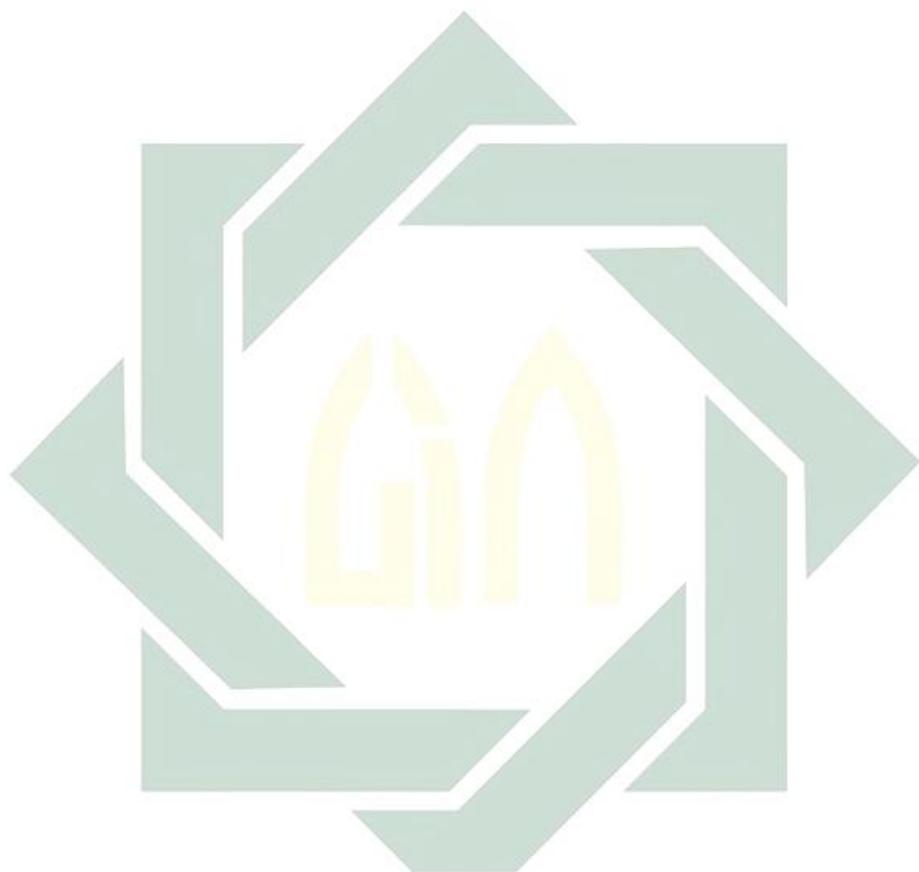
Gambar 4.6 Citra dengan Nilai PSNR Rendah



Gambar 4.7 Citra Prediksi yang Mempunyai Kemiripan dengan Citra Aslinya

Keberhasilan DeepESN untuk memprediksi kondisi awan pada jam berikutnya ditunjukkan pada Gambar 4.7 bagian kanan. Gambar asli yang

seharusnya didapatkan ditunjukkan pada Gambar 4.7 bagian kanan. Kedua Gambar tersebut memiliki kemiripan di daerah yang berawan dan tidak berawan. Perbedaannya yaitu citra hasil prediksi memiliki tampilan yang lebih halus jika dibandingkan dengan citra aslinya.



BAB V

PENUTUP

Pada bab ini akan diberikan simpulan dan saran-saran yang dapat diambil berdasarkan materi-materi yang telah dibahas pada bab-bab sebelumnya.

5.1. Simpulan

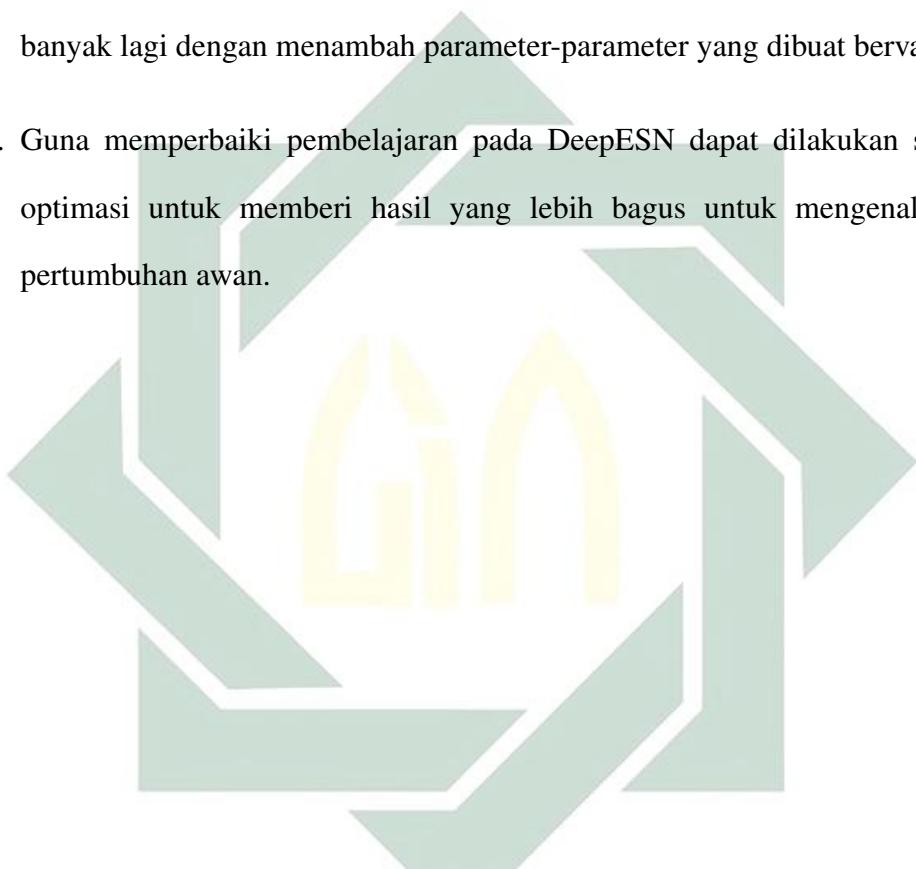
Melihat dari hasil dan pembahasan peniliti dapat mengambil beberapa kesimpulan yaitu ;

1. Prediksi awan cumulonimbus pada citra himawari dapat dipelajari menggunakan DeepESN. Prediksi menggunakan DeepESN diperlukan beberapa uji coba untuk mendapatkan struktur yang bagus. Pada penelitian ini struktur yang paling bagus menggunakan *learning rate* 0.5, jumlah node 120, dan lapisan *reservoir* sebanyak 150 lapisan. Proses prediksi dilakukan dengan menngambil citra merah dan menjadikan satu citra dalam sebuah vektor. Setiap vektor citra dimasukkan dala sistem DeepESN untuk dipelajari pola pertumbuhan awannya.
 2. Pada penelitian ini pengukuran keberhasilan dilihat dari tingkat kemiripan antara citra sebenarnya dengan citra hasil prediksi. Penilaian rata-rata kemiripan citra dengan metode PSNR didapatkan hasil terbaik yaitu 22.994. Rata-rata PSNR tersebut memiliki nilai PSNR terkecil yaitu 8.676, dan nilai kemiripan tertinggi yaitu 51.2006 dB.

5.2. Saran

Melihat dari beberapa kekurangan yang ada pada penelitian ini, peneliti ingin memberikan saran untuk penelitian selanjutnya.

- 
 1. Prediksi menggunakan DeepESN dapat diberikan pengujian yang lebih banyak lagi dengan menambah parameter-parameter yang dibuat bervariasi.
 2. Guna memperbaiki pembelajaran pada DeepESN dapat dilakukan sebuah optimasi untuk memberi hasil yang lebih bagus untuk mengenali pola pertumbuhan awan.



DAFTAR PUSTAKA

BMKG. (2019). *Citra Satelit*. Retrieved October 12, 2019, from
<https://www.bmkg.go.id/satelite/satelite.bmkg?Sat=6&id=0>

Chen, H.-L., Yang, B., Liu, J., & Liu, D.-Y. (2011). *A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis*. Expert Systems with Applications, 38(7), 9014–9022.

Dewi, C., & Supianto, A. A. (2015). *Pengolahan Citra Satelit Dengan MATLAB*. Universitas Brawijaya Press.

Efendi, A. N., & Fachrurrozi, M. (2017). *Pemanfaatan Data Satelit Ir 1 Himawari-8 untuk Mengurangi Dampak Kerugian Materiil dan Jiwa Akibat Bencana Alam di Wilayah Melawi Kalimantan Barat* The Utilization of Ir 1 Himawari-8 Satellite Data to Reduce the Impact of Material Loss and Soul Due to Natu.

Emiliani, C. (1992). *Planet earth: cosmology, geology, and the evolution of life and environment*. Cambridge University Press.

Gallicchio, C., & Micheli, A. (2017). *Deep Echo State Network (DeepESN): A Brief Survey*. 1–13. Retrieved from <http://arxiv.org/abs/1712.04323>

Gallicchio, C., & Micheli, A. (2018). *Experimental analysis of deep echo state networks for ambient assisted living*. CEUR Workshop Proceedings, 2061, 44–57.

Gallicchio, C., Micheli, A., & Pedrelli, L. (2018). *Design*

<https://doi.org/10.1016/j.neunet.2018.08.002>

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). *Draw: A recurrent neural network for image generation*. ArXiv Preprint ArXiv:1502.04623.

Handoyo, S. (2011). *Aviapedia: ensiklopedia umum penerbangan* (Vol. 1). Penerbit Buku Kompas.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). *A practical guide to support vector classification*.

Imam, M. P. B., & Pamungkas, W. (2014). *Sistem Komunikasi Satelit*. Penerbit Andi.

Kadir, A., & Susanto, A. (2013). *Teori Dan Aplikasi Pengolahan Citra*, Yogyakarta.
Andi Publisher.

Klibisz, A. (2016). *A Primer on Reservoir Computing*.

Koryakin, D., & Butz, M. V. (2012). *Reservoir sizes and feedback weights interact non-linearly in echo state networks*. International Conference on Artificial Neural Networks, 499–506. Springer.

Lacy, S. E., Smith, S. L., & Lones, M. A. (2018). *Using echo state networks for classification: A case study in Parkinson's disease diagnosis*. Artificial Intelligence in Medicine, 86, 53–59.
<https://doi.org/10.1016/j.artmed.2018.02.002>

Lee, C. S., Baughman, D. M., & Lee, A. Y. (2017). *Deep Learning Is Effective for Classifying Normal versus Age-Related Macular*

- Degeneration OCT Images.* Kidney International Reports, 2(4), 322–327.
<https://doi.org/10.1016/j.oret.2016.12.009>
- Lee, Y., & Lee, C.-K. (2003). *Classification of multiple cancer types by multiclass support vector machines using gene expression data.* Bioinformatics, 19(9), 1132–1139.
- Liu, X., Chen, M., Yin, C., & Saad, W. (2018). *Analysis of Memory Capacity for Deep Echo State Networks.* 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 443–448. IEEE.
- Long, J., Zhang, S., & Li, C. (2019). *Evolving Deep Echo State Networks for Intelligent Fault Diagnosis.* IEEE Transactions on Industrial Informatics.
- Lukocevicius, M., & Jaeger, H. (2009). *Reservoir computing approaches to recurrent neural network training.* Computer Science Review, 3(3), 127–149.
- Ma, Q., Shen, L., Chen, W., Wang, J., Wei, J., & Yu, Z. (2016). *Functional echo state network for time series classification.* Information Sciences, 373, 1–20.
<https://doi.org/10.1016/j.ins.2016.08.081>
- Magnin, B., Mesrob, L., Kinkingnehun, S., Pelegrini-Issac, M., Colliot, O., Sarazin, M., ... Benali, H. (2009). *Support vector machine-based classification of Alzheimer's disease from whole-brain anatomical MRI.* Neuroradiology, 51(2), 73–83.
- Massinai, M. A. (2005). *Analisis Liputan Awan Berdasarkan Citra Satelit Penginderaan Jauh.* Pertemuan Ilmiah Masyarakat Penginderaan Jauh XIV. ITS.
- Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., & Khudanpur, S. (2010).

Recurrent neural network based language model. Eleventh Annual Conference of the International Speech Communication Association.

Nugraheny, D. (2015). *Metode Nilai Jarak guna Kesamaan atau Kemiripan Ciri suatu Citra (kasus deteksi awan cumulonimbus menggunakan principal component analysis)*. Angkasa: Jurnal Ilmiah Bidang Teknologi, 7(2), 21–30.

Pandjaitan, B. S., & Damayanti, M. I. (2018). *Kajian Mikrofisis Awan Menggunakan Satelit Himawari 8 pada Kejadian Hail (Studi Kasus: Kejadian Hail di Jakarta Tanggal 28 Maret 2017) Study of Cloud Microphysic Using Himawari 8 Satellite on the Hail Stone Event (Case Study: Hail Stone Event in Jakarta o.*

Paski, J. A. I., Permana, D. S., Sepriando, A., & Pertiwi, D. A. S. (2017). *Analisis Dinamika Atmosfer Kejadian Hujan Es Memanfaatkan Citra Radar dan Satelit Himawari-8 (Studi Kasus: Tanggal 3 Mei 2017 di Kota Bandung) Atmospheric Dynamics Analysis of Hail Event Utilizing Radar and Himawari-8 Satellite Imagery (Case Study: May 3, 2017 in Bandung City)*.

PirahanSiah, F., Abdullah, S. N. H. S., & Sahran, S. (2010). *Adaptive image segmentation based on peak signal-to-noise ratio for a license plate recognition system*. In 2010 International Conference on Computer Applications and Industrial Electronics (pp. 468–472). IEEE.

Putra, D. (2010). *Pengolahan citra digital*. Penerbit Andi.

Roza, E. (2019). *Maritim Indonesia, Kemewahan Yang Luar Biasa*. Retrieved October 12, 2019, from <https://kkp.go.id/artikel/2233-maritim-indonesia-kemewahan-yang-luar-biasa>

- Schaetti, N., Salomon, M., & Couturier, R. (2016). *Echo state networks-based reservoir computing for mnist handwritten digits recognition*. 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), 484–491. IEEE.

Schrauwen, B., Verstraeten, D., & Van Campenhout, J. (2007). *An overview of reservoir computing: theory, applications and implementations*. Proceedings of the 15th European Symposium on Artificial Neural Networks. p. 471-482 2007, 471–482.

Suwargana, N. (2013). *Resolusi spasial, temporal dan spektral pada citra satelit Landsat, SPOT dan IKONOS*. Jurnal Ilmiah Widya, 1(2), 167–174.

Utoyo, B. (2007). *Geografi: Membuka Cakrawala Dunia*. PT Grafindo Media Pratama.

Van der Sande, G., Brunner, D., & Soriano, M. C. (2017). *Advances in photonic reservoir computing*. Nanophotonics, 6(3), 561–576.

Wolf, S., & Pinson, M. H. (2009). *Reference algorithm for computing peak signal to noise ratio (psnr) of a video sequence with a constant delay*. ITU-T Contribution COM9-C6-E.

Yao, S., Lin, W., Ong, E., & Lu, Z. (2005). *Contrast signal-to-noise ratio for image quality assessment*. In IEEE International Conference on Image Processing 2005 (Vol. 1, pp. I–397). IEEE.

Yusoff, M.-H., & Jin, Y. (2014). *Modeling neural plasticity in echo state*

networks for time series prediction. 2014 14th UK Workshop on Computational Intelligence (UKCI), 1–7. IEEE.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). *Recurrent neural network regularization*. ArXiv Preprint ArXiv:1409.2329.