

**PREDIKSI CUACA BERDASARKAN KORELASI
MULTI-PARAMETER FENOMENA FISIS ALAM
DENGAN *MACHINE LEARNING***

SKRIPSI

Oleh :

Ragil Bagus Agung Budiyono

175090307111003



**JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU
PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2021**

LEMBAR PENGESAHAN SKRIPSI
PREDIKSI CUACA BERDASARKAN KORELASI MULTI-
PARAMETER FENOMENA FISIS ALAM DENGAN
MACHINE LEARNING

Oleh:

Ragil Bagus Agung Budiyo
175090307111003

PROGRAM STUDI: S1 FISIKA
Malang, Maret 2021

Pembimbing I

Pembimbing II

(Agus Naba, S.Si., MT., Ph.D)
NIP. 197208061995121001

(Dr.rer.nat. Abdurrouf, S.Si.,M.Si)
NIP. 197209031994121001

Mengetahui,

Ketua Program Studi S1 Fisika
Jurusan Fisika FMIPA UB

(Dr.Eng Masruroh, S.Si.,M.Si.)
NIP. 197512312002122002

KATA PENGANTAR

Puji syukur kepada Allah SWT, karena atas segala limpahan rahmat, rezeki serta nikmat-Nya penulis dapat menyelesaikan proposal Tugas Akhir dengan judul **“PREDIKSI CUACA BERDASARKAN KORELASI MULTI-PARAMETER FENOMENA FISIS ALAM DENGAN MACHINE LEARNING”**.

Proposal Tugas Akhir ini bertujuan sebagai gambaran awal mengenai Tugas Akhir yang akan dilaksanakan dan untuk memenuhi syarat melakukan ujian Tugas Akhir selanjutnya.

Dalam penyusunan proposal Tugas Akhir ini, penulis memberikan rasa terima kasih kepada pihak yang telah memberikan bantuan, dukungan, bimbingan serta saran sampai penulis dapat menyelesaikan proposal ini, khususnya kepada:

1. Allah SWT atas nikmat, rahmat, dan rezeki sehingga penulisan Proposal Tugas Akhir dapat diselesaikan dengan baik.
2. Kedua orang tua yang selalu memberikan dukungan baik moral maupun material.
3. Bapak Prof. Dr. rer.nat Muhammad Nurhuda. selaku Ketua Jurusan Fisika FMIPA Universitas Brawijaya Malang serta dosen pembimbing akademik.
4. Ibu Dr. Eng. Masruroh, M.Si. selaku ketua prodi Fisika.
5. Bapak Agus Naba, S.Si., MT., Ph.D selaku dosen Pembimbing I.
6. Bapak Dr.rer.nat. Abdurrouf, S.Si., M.Si selaku dosen Pembimbing I.
7. Teman-teman Jurusan Fisika FMIPA Universitas Brawijaya Malang, sahabat dekat serta pihak yang

tidak dapat disebutkan yang telah memberikan semangat, dorongan serta motivasi.

Penulis mengetahui bahwa karya selalu ada cela, sehingga dari pembuatan karya tulis ini apabila terdapat kritik, saran serta masukan dapat diberikan ke penulis untuk mengembangkan dan memperbaiki dari karya tulis ini maupun diri penulis sendiri.

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Manfaat.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 Cuaca.....	5
2.1.1 Parameter Cuaca.....	6
2.1.2 Fenomena Fisis Cuaca	7
2.2 Artificial Intelligence	8
2.2.1 Machine Learning.....	9
2.2.2 Supervised Learning	10
2.2.3 Regresi dan Klasifikasi	11
2.2.4 Neural Network.....	12
2.2.5 Time-Series Neural Network	14
2.2.6 K-Nearest Neighbors	16
2.2.7 Loss Function.....	17

2.3	Python.....	18
2.3.1	TensorFlow	18
2.3.2	Keras	19
2.3.3	Scikit-Learn	19
BAB III METODOLOGI.....		20
3.1	Waktu dan Tempat Pelaksanaan.....	20
3.2	Alat dan Bahan	20
3.3	Tahapan Penelitian.....	20
3.3.1	Persiapan Komputasi	21
3.3.2	Pengolahan Data.....	22
3.3.3	Pembuatan Model Neural Network dan K-Nearest Neighbors.....	22
3.3.4	Pelatihan Neural Network dan K-Nearest Neighbors	24
3.3.5	Pengujian Neural Network dan K-Nearest Neighbors	24
3.3.6	Evaluasi dan Visualisasi ML	25
BAB IV JADWAL KEGIATAN PENELITIAN.....		26
4.1	Jadwal Kegiatan Penelitian	26
DAFTAR PUSTAKA.....		27

DAFTAR GAMBAR

Gambar 2.2.1. Gambar yang merujuk deep learning masuk cakupan machine learning, dan machine learning masuk cakupan AI.....	9
Gambar 2.2.2. Alur kerja sistem pembelajaran supervised learning.....	11
Gambar 2.2.3. Plot kiri merupakan algoritma klasifikasi dan kanan untuk regresi.....	12
Gambar 2.2.4. Gambaran Neural Network.....	13
Gambar 2.2.5. Skema LSTM dan unit memorinya.....	16
Gambar 2.2.6. K-Nearest Neighborss untuk data banyaknya gaji bulanan tiap pekerja.....	17
Gambar 3.3.1. Arsitektur LSTM	23

BAB I

PENDAHULUAN

1.1 Latar Belakang

Cuaca sangat mempengaruhi bagaimana manusia menjalankan kegiatannya, menjadikan cuaca sebagai faktor yang harus dipertimbangkan dalam berbagai sektor. Pada pertanian, cuaca mampu mempengaruhi bagaimana tiap tanaman dapat bertumbuh dengan optimal, Parameter dari cuaca sangat berpengaruh untuk pertumbuhan tanaman seperti kelembapan, temperatur dan lainnya (Cogato et al., 2019). Cuaca juga dapat mempengaruhi bagaimana sektor perairan, bukan hanya pertanian, cuaca mempengaruhi pasokan air (Kirono et al., 2016). Bahkan cuaca juga mempengaruhi bagaimana sebuah pandemi dikendalikan, dengan melihat parameter cuaca maka transmisi dari pandemi dapat diprediksi dan memberikan tindak lanjut yang tepat (Tosepu et al., 2020).

Pentingnya cuaca dalam kehidupan manusia yang melibatkan berbagai sektor mendorong manusia untuk melakukan peramalan dan prediksi cuaca, sehingga dampak dari cuaca dapat diantisipasi bahkan sebelum cuaca tersebut terjadi. Prediksi cuaca umumnya melibatkan satu parameter yang nantinya akan dikorelasikan dengan dampak dari parameter tersebut. Seperti yang ditulis oleh Khosravi dan kawan-kawan, prediksi kecepatan angin dilakukan untuk melihat dampak kecepatan angin ke *wind farm* atau pembangkit listrik tenaga angin (Khosravi et al., 2018). Prediksi dengan satu parameter juga telah dilakukan oleh banyak peneliti, seperti prediksi curah hujan dan temperatur dengan fungsi bergantung pada tujuan penulis (Kirono et al., 2016; Volokitin et al., 2016).

Peramalan dengan satu parameter dapat dilakukan dengan tujuan spesifik, karena cuaca secara umum merupakan masalah multi parameter sehingga banyak hal yang dapat mempengaruhinya. Tanpa disertai dengan peramalan, terdapat penelitian yang membahas tentang klasifikasi cuaca berdasarkan data yang ada. Terdapat penelitian yang memberikan klasifikasi cuaca berdasarkan foto yang diambil pada saat itu dan algoritma yang dijalankan akan mengidentifikasi tergolong apakah cuaca pada hari itu (Zhang et al., 2016). Penggolongan cuaca memang bermanfaat dalam beberapa bidang, seperti penggunaannya untuk mengidentifikasi cuaca dari kamera sehingga tidak diperlukannya manusia untuk mengetahui cuaca secara manual, tetapi hanya menggolongkan cuaca tidak dapat memberikan prediksi cuaca kedepannya.

Berdasarkan latar belakang tersebut, penulis akan membangun sebuah algoritma *time-series machine learning* sehingga prediksi tiap parameter cuaca akan dilakukan dan hasil prediksi tersebut digolongkan menggunakan algoritma *classification machine learning*, sehingga didapatkan ramalan cuaca multi parameter. Dalam penelitian kedepannya juga akan diteliti terkait hubungan antar parameter fisis yang paling mempengaruhi sehingga tercipta kategori cuaca pada waktu tersebut.

1.2 Rumusan Masalah

Dari latar belakang tersebut, maka diperoleh beberapa rumusan masalah sebagai berikut:

1. Bagaimana korelasi antara parameter fisis (Suhu minimum-maksimum, kecepatan dan arah angin, presipitasi dan kelembapan) untuk meramalkan cuaca?

2. Bagaimana cara memprediksi cuaca pada rentang jam dengan durasi mingguan menggunakan multi-parameter?
3. Bagaimana keakuratan model yang dibuat dan korelasinya terhadap masing-masing parameter fisis apabila parameter ditambahkan atau dikurangi?
4. Bagaimana kemampuan model untuk memprediksi cuaca dengan berbagai rentang waktu dan parameter yang digunakan?

1.3 Tujuan

Dari latar belakang dan rumusan masalah tersebut, maka diperoleh tujuan penelitian sebagai berikut:

1. Mengetahui korelasi antara parameter fenomena fisis terhadap cuaca yang terjadi.
2. Mengetahui cara untuk memprediksi cuaca terhadap multi-parameter fenomena fisis dengan metode regresi dan klasifikasi.
3. Mengetahui kemampuan model terhadap korelasi tiap parameter dan perubahan hasil apabila parameter ditambahkan atau dikurangi.
4. Mengetahui kemampuan model terhadap prediksi tahunan pada rentang waktu dan parameter terlibat yang ditentukan.

1.4 Batasan Masalah

Batasan masalah yang diberikan pada penelitian yang akan dilakukan adalah sebagai berikut:

1. Cuaca yang diprediksi memiliki kriteria cerah, Sebagian mendung, mendung, hujan, cerah disertai hujan, sebagian mendung disertai hujan

2. Cuaca yang diprediksi berdasarkan data curah hujan tiap jam di kota Malang pada Januari 2010 – Januari 2021
3. Parameter faktor penentu hujan secara fisis yang digunakan adalah suhu minimum-maksimum, kecepatan dan arah angin, presipitasi dan kelembapan.

1.5 Manfaat

Penelitian ini memiliki beberapa manfaat yaitu:

1. Bagi Instansi terkait: Hasil model penelitian dapat digunakan sebagai alat prediksi cuaca pertama sebelum menggunakan alat dengan akurasi tinggi
2. Bagi Peneliti dibidang Iklim dan Cuaca: Hasil model penelitian dapat digunakan untuk membuat dan menganalisis prediksi cuaca pada skala mingguan dengan ketelitian per jam

BAB II

TINJAUAN PUSTAKA

2.1 Cuaca

Cuaca memiliki makna yang bercampur dengan iklim, sehingga perbedaan arti dari istilah tersebut menjadi kabur. Padahal kedua istilah tersebut memiliki makna yang cukup berbeda apabila ditinjau dari segi waktu. Cuaca sendiri merupakan keadaan atmosfer pada tiap satuan waktu, sehingga terdapat rujukan waktu yang dituju. Penggunaan cuaca dengan konotasi waktu seperti cuaca hari ini, cuaca besok, cuaca jam 1 siang dan sebagainya pasti mengarah pada waktu tertentu untuk mengutarakan keadaan atmosfer.

Cuaca memiliki berbagai parameter yang diungkapkan seperti suhu, tekanan, angin, kelembapan yang diukur dengan alat khusus. Parameter tersebut akan mampu memberikan probabilitas keadaan yang terjadi, seperti akan turun hujan, mendung, cerah dan sebagainya. Fenomena fisis sangat berpengaruh pada keadaan cuaca, karena cuaca dibentuk dari berbagai faktor yang terjadi di atmosfer seperti rotasi bumi, kelembapan, suhu dikarenakan intensitas sinar matahari dan lain sebagainya (Wirjohamidjojo & Swarinoto, 2010).

Pentingnya cuaca juga selaras dengan pentingnya prediksi cuaca. Peramalan cuaca sulit dilakukan karena cuaca memiliki banyak faktor, bukan hanya parameter cuaca saja yang diperhatikan tapi bagaimana posisi bumi terhadap matahari dan bulan, juga pengaruh alam yang lain. Banyak metode prediksi yang dilakukan untuk meningkatkan upaya keberhasilan dalam prediksi itu sendiri, mulai dengan perhitungan secara manual dengan

memperhitungkan parameter fisis, menggunakan regresi maupun menggunakan *machine learning* (Watts, 2014).

2.1.1 Parameter Cuaca

Cuaca memiliki parameter yang akan menggambarkan kondisi dari waktu tertentu, sehingga tiap parameter akan menggambarkan keadaan cuaca berdasarkan nilai dari parameter dan korelasi dari keseluruhan parameter. Parameter dari cuaca sendiri diukur dengan berbagai macam alat dan kondisi tertentu (Wirjohamidjojo & Swarinoto, 2010).

Beberapa parameter cuaca yang dapat dilihat adalah suhu minimum, suhu maksimum, keadaan angin, keadaan awan dan presipitasi, kondensasi air, dan evaporasi. Suhu minimum didapatkan dari pengukuran suhu pada malam hari dengan memperhitungkan keadaan berawan. Suhu minimum penting untuk menjaga tanaman tidak rusak karena keluar dari zona tumbuh. Suhu maksimum didapatkan dari keadaan cerah di siang hari, makna suhu maksimum sebenarnya adalah menandakan suhu maksimum dari udara. Keadaan angin didapatkan dari pengukuran kecepatan angin dan arah angin. Angin timbul akibat perubahan tekanan dan suhu pada suatu wilayah. Keadaan awan dan presipitasi dapat mendeskripsikan keadaan berawan atau hujan, dengan jumlah air di udara maupun di atmosfer diperhitungkan. Evaporasi merupakan parameter yang timbul akibat parameter lain, yaitu suhu juga luas permukaan maupun lokasi pengamatan, evaporasi dapat memberikan pandangan terkait siklus air yang nantinya akan membentuk awan dan sebagainya (Potter & Coleman, 2003).

2.1.2 Fenomena Fisis Cuaca

Bagaimana cuaca dan parameternya timbul tentu tidak akan jauh dari fisika karena cuaca merupakan produk dari fenomena fisika yang berada di alam. Mengerti bagaimana fenomena fisis terjadi di alam terkait dengan timbulnya nilai di parameter cuaca dapat menjelaskan mengenai parameter itu sendiri dan bisa didapatkannya metode untuk prediksi secara manual. Walau ternyata prediksi menggunakan cara manual ternyata menghasilkan hasil yang kurang baik, memahami korelasi tiap parameter dapat meningkatkan hasil prediksi pada metode yang lain (Lions et al., 1992).

Fenomena fisis yang terjadi dari tiap parameter timbul dari intensitas radiasi matahari, kedekatan jarak bumi-matahari, perubahan kecepatan dan arah angin akibat perbedaan suhu dan tekanan, timbulnya hujan deras dengan berbagai jenis hujan dikarenakan bertambahnya uap air di awan yang turun disertai gesekan udara pada saat itu, bagaimana transfer energi yang terjadi dari sinar matahari ke bumi yang kemudian akan meningkatkan evaporasi, massa jenis uap air dengan masa jenis atmosfer saat itu dan lain sebagainya. Apabila semua faktor fisis dimasukkan kedalam prediksi maka akan timbul tingkat kekompleksan yang tinggi dan juga akan memakan sumber daya komputasi yang tinggi, sehingga faktor fisis biasanya disederhanakan menjadi beberapa bagian saja (Wirjohamidjojo & Swarinoto, 2010).

Variabel utama yang diambil untuk fenomena fisis adalah volume udara atau air di atmosfer, massa jenis, temperatur, dan tekanan. Dari variabel tersebut umumnya para peneliti cuaca (meteorologis) menggunakan beberapa

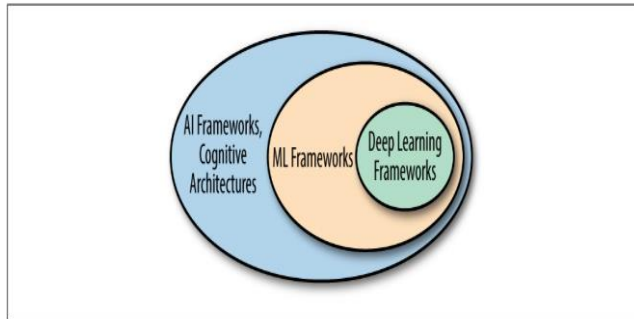
model matematis untuk menjelaskan fenomena fisis tersebut, diantaranya persamaan konservasi momentum Navier-Stokes, hukum pertama termodinamika terkait konservasi energi, persamaan konservasi massa udara, persamaan kontinuitas massa uap air dan persamaan gas ideal. Persamaan tersebut dapat menjadi acuan korelasi antar parameter, tetapi korelasi antar parameter pastinya beragam bergantung pada faktor lain, dan hal tersebut menyebabkan prediksi menggunakan persamaan matematis memiliki hasil yang tidak optimal (Lions et al., 1992; Potter & Coleman, 2003).

2.2 Artificial Intelligence

Artificial Intelligence (AI) secara langsung berarti kecerdasan buatan, merujuk pada kemampuan komputer dalam menganalisis sebuah permasalahan dengan cerdas. AI sendiri tidak bermakna sempit merujuk pada sebuah robot cerdas, tapi sebuah algoritma yang mampu menemukan pola tertentu yang terdapat pada data. AI dapat diaplikasikan pada *e-commerce*, *facial recognition*, dan lainnya. AI memiliki beberapa cabang seperti *Machine Learning*, *Natural Language*, *Fuzzy Logic*, dan lain sebagainya (Lu et al., 2017).

Cakupan AI sangat luas, mencakup penyelesaian regresi, klasifikasi, *clustering* hingga ke penalaran, perencanaan dan navigasi. Ide besar dari AI adalah mampu menyelesaikan kebutuhan manusia yang mendasar dengan baik atau bahkan *lebih* baik. Walaupun usaha untuk mencapai hal tersebut sangat sulit karena otak manusia terdiri dari jutaan syaraf yang terhubung dan memiliki fungsionalitas masing-masing, AI merupakan

terobosan maju di bidang teknologi. Cakupan AI dan *machine learning* maupun *deep learning* dapat dilihat pada Gambar 2.2.1. Gambar yang merujuk deep learning masuk cakupan machine learning, dan machine learning masuk cakupan AI (Morgan, 2018).



Gambar 2.2.1. Gambar yang merujuk deep learning masuk cakupan machine learning, dan machine learning masuk cakupan AI.

2.2.1 Machine Learning

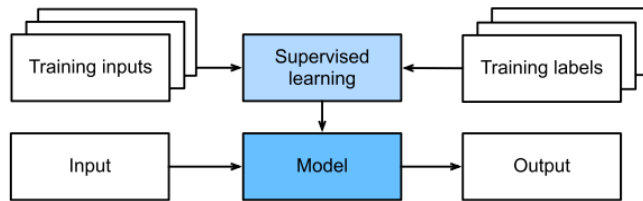
Machine Learning (ML) merupakan algoritma naungan AI. Fungsionalitas awalnya adalah mengubah data mentah menjadi sesuatu yang bernilai, hal tersebut berlaku pada ML secara konvensional. Dengan penelitian yang matang, ML pada terobosan terbarunya mampu benar-benar mengekstrak sebuah pola pada data mentah dalam berbagai jenis data. Pola yang diambil dari data mentah dapat berupa konstanta yang mempengaruhi tiap parameter, sehingga dari tiap data dapat ditemukan pola regresinya dan melengkapi atau meramalkan data yang belum ada, bahkan mampu mengklasifikasikan data pada

pola tertentu bergantung dengan hasil yang didapatkan setelah proses *train* dilakukan.

ML merupakan pembahasan yang hangat karena kemampuan menemukan pola dari sebuah data mentah, sehingga proses *engineering* kode program menjadi lebih efisien, karena dengan memberikan algoritma tertentu kode program dapat beradaptasi dengan baik tanpa harus memberikan perlakuan khusus secara manual oleh pengguna. Terlebih pada era informasi saat ini, data merupakan hal yang mudah didapatkan dari berbagai sumber dan data adalah bahan utama agar ML bisa bekerja dengan baik (Lecun et al., 2015).

2.2.2 Supervised Learning

Supervised learning adalah sistem pembelajaran di ML dengan memberikan kategori label di tiap keluarannya. Pemberian kategori label yang dimaksud adalah hasil keluaran yang diharapkan sudah ditentukan, misal regresi dengan label curah hujan atau klasifikasi dengan label membedakan motor dan mobil. *Supervised learning* sangat umum digunakan karena penggunaannya yang sangat luas, berbeda dengan sistem lain seperti *unsupervised learning* yang tidak memberikan label pada keluaran sehingga ML harus menentukan labelnya sendiri berdasarkan data (Lecun et al., 2015).



Gambar 2.2.2. Alur kerja sistem pembelajaran supervised learning.

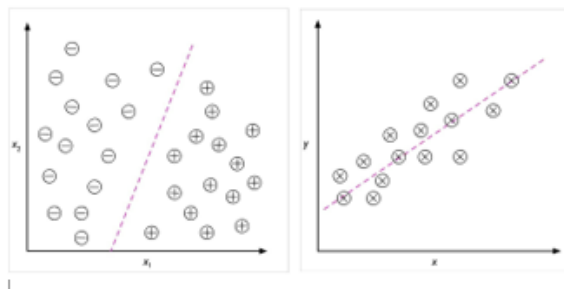
Alur pengerjaan *supervised learning* dapat dilihat seperti pada Gambar 2.2.2. Alur kerja sistem pembelajaran supervised learning. Data berupa *training input* disiapkan sebagai bahan pelatih dari ML untuk menghasilkan model. *Training label* juga ditentukan untuk memberikan informasi ke ML keluaran yang seharusnya. Apabila keduanya telah disiapkan, maka data akan di-*train* kemudian menghasilkan model yang tepat berdasarkan data masukan. Model yang dihasilkan dapat diuji maupun digunakan dengan *input test* dan akan menghasilkan keluaran berdasarkan model yang telah di-*train* sebelumnya (Czum, 2020).

2.2.3 Regresi dan Klasifikasi

Supervised learning dapat melakukan berbagai macam hal, pada umumnya pengguna akan menggunakan *supervised learning* sebagai sistem pembelajaran untuk menyelesaikan regresi dan klasifikasi. Sederhananya regresi merupakan prediksi yang dilakukan dengan angka real berdasarkan data yang ada. Regresi dapat menambahkan data yang kosong maupun meneruskan data yang sudah ada. Contoh regresi adalah prediksi harga

rumah tahunan, prediksi harga saham, prediksi curah hujan. Kunci dari regresi adalah keluaran yang berupa nilai kontinu, sehingga sangat cocok untuk menyelesaikan permasalahan prediksi.

Lain halnya klasifikasi, ML akan mengelompokkan data sesuai dengan kategori label yang telah ditentukan (pada kasus *supervised learning*). Keluaran dari klasifikasi adalah nilai diskrit yang menentukan label dari data. Contohnya adalah bagaimana klasifikasi kucing dan anjing, kanker ganas dan tidak, serta klasifikasi keadaan cuaca (Czum, 2020). Perbedaan klasifikasi dan regresi dapat dibandingkan di Gambar 2.2.3. Plot kiri merupakan algoritma klasifikasi dan kanan untuk regresi.. Klasifikasi digunakan untuk mengelompokkan dan regresi untuk menemukan pola data (Raschka & Mirjalili, 2017).

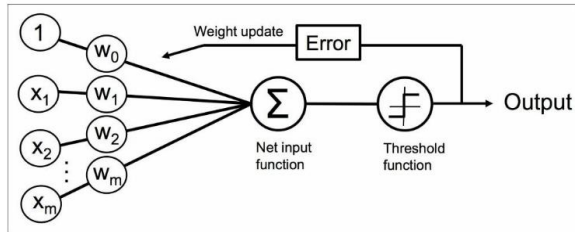


Gambar 2.2.3. Plot kiri merupakan algoritma klasifikasi dan kanan untuk regresi.

2.2.4 Neural Network

Algoritma ML yang cukup terkenal sebagai *powerhouse* untuk menyelesaikan banyak permasalahan adalah *Neural Network* (NN). NN dibangun berdasarkan penggambaran sistem syaraf manusia, ketika beragam

input masuk lalu diproses untuk menghasilkan keluaran biner. NN akan mempelajari data yang diberikan lalu memberikan *update* bobot sehingga model akan semakin baik, maka dari itu NN merupakan algoritma yang dilakukan secara berulang (*epoch*) untuk menentukan bobot yang sesuai dengan label keluaran.



Gambar 2.2.4. Gambaran Neural Network.

Skema NN seperti pada Gambar 2.2.4. Gambaran Neural Network. menjelaskan terkait bagaimana *input* (x) yang diberikan akan diproses dengan bobot awal untuk membuat model berupa fungsi. Data kemudian diproses dengan *activation function* yang menjadikan hasil menjadi keluaran diskrit (sesuai dengan jenis *activation function*) lalu dievaluasi dengan fungsi *error* (atau *loss function*). Apabila sudah sesuai dengan batas maka training dihentikan, apabila belum maka hasil dari *error* tersebut digunakan untuk memberikan *update* ke bobot sehingga *epoch* selanjutnya akan menghasilkan *error* yang lebih kecil.

Model matematis sederhana untuk menggambarkan *update* bobot adalah sebagai berikut:

$$w_j := w_j + \Delta w_j \quad (2.1)$$

$$\Delta w_j = \eta (y^{(i)} - \hat{y}^{(i)}) x_j^{(i)} \quad (2.2)$$

w_j menggambarkan bobot pada unit j , akan di-*update* setiap kali ada perubahan w_j yang ditandai dengan Δw_j . Δw_j adalah kalkulasi yang memperhitungkan η sebagai *learning rate* (ditentukan sendiri oleh pengguna) untuk menunjukkan besarnya langkah yang perlu diambil tiap bobot di-*update* (semakin kecil semakin presisi tetapi memakan sumber daya komputasi dan waktu), $y^{(i)}$ sebagai label keluaran atau target *output*, $\hat{y}^{(i)}$ adalah keluaran *training* dengan bobot sebelumnya, dan $x_j^{(i)}$ adalah input. Persamaan matematis tersebut dapat diartikan bahwa delta bobot (perubahan bobot) didapatkan dari selisih hasil keluaran *training* dengan data sesungguhnya per data *train* $x_j^{(i)}$, dengan kontrol step dikendalikan oleh *learning rate* (Raschka & Mirjalili, 2017).

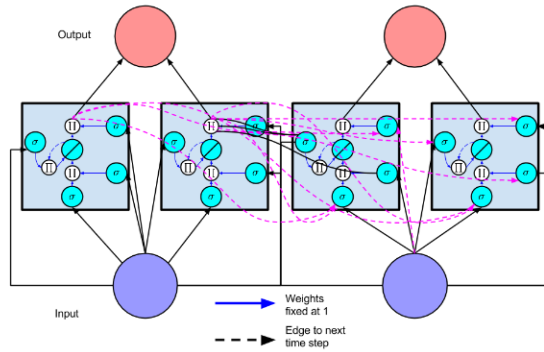
2.2.5 Time-Series Neural Network

NN punya kecenderungan untuk menemukan pola pada sebuah data masukan. Penemuan pola tersebut didasarkan pada *input* sebelumnya, sehingga data yang telah di proses akan mempengaruhi bobot ke depannya. Permasalahannya adalah NN konvensional cenderung “lupa” apabila data yang dipanggil terlalu lama, dan NN konvensional tidak sensitif terhadap waktu. Karena itu, NN konvensional kurang baik untuk mengelola data dengan waktu sensitif seperti ramalan curah hujan dan harga saham. Permasalahan tersebut dapat diselesaikan dengan beberapa metode, yaitu pembuatan *time-series*

neural network. Banyak metode pembuatan *time-series* NN, seperti *recurrent NN*, dan yang terbaru adalah LSTM.

Recurrent NN sebenarnya adalah modifikasi dari *feed-forward Neural Network* dengan menambahkan komponen waktu. *Recurrent NN* mampu mengingat data yang telah lampau dengan urutan yang baik. Kelemahannya adalah *Recurrent NN* tidak dapat memilih konstanta mana yang harus diingat dan dilupakan, sehingga muncul metode baru yaitu *Long-Short Term Memory* (LSTM).

LSTM memiliki kelebihan yaitu adanya unit tambahan yang berfungsi untuk mengurutkan data sesuai dengan urutan kejadian waktu dan mengingatnya (seperti unit memori). Unit tersebut memiliki *gate* khusus dengan dimulai dari nilai *Input Node* (**g**), *Input Gate* (**i**), *Forget Gate* (**f**), *Output Gate* (**o**). *Gate* akan menentukan memori mana yang harus dipertahankan atau dilupakan, yang akan mempengaruhi perhitungan dari bobot. Kita lihat dari gambar ada 3 *gate* (lingkaran berwarna putih), dengan *gate* pertama yaitu *Input Gate* untuk memberikan perubahan ke data input, lalu *Forget Gate* akan menentukan apakah ada konstanta yang harus dilupakan, apabila ada maka akan dihapus dan digantikan pada *Output Gate* (Lipton et al., 2015).



Gambar 2.2.5. Skema LSTM dan unit memorinya

2.2.6 K-Nearest Neighbors

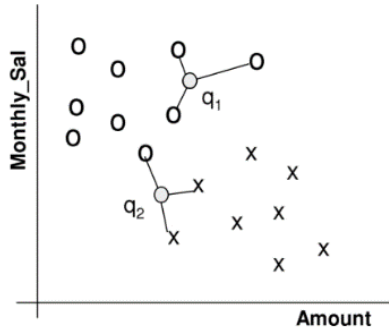
Metode klasifikasi memiliki banyak bentuk, *K-Nearest Neighbors* memanfaatkan data sekitarnya untuk mengklasifikasikan data yang ditunjuk. Apabila kita lihat pada Gambar 2.2.6. K-Nearest Neighbors untuk data banyaknya gaji bulanan tiap pekerja., q_1 dan q_2 adalah data yang ingin diklasifikasikan. Data yang ingin diklasifikasikan tersebut diukur jaraknya dengan beberapa *tetangga terdekat* yang telah diketahui labelnya. Dari pengukuran jarak terhadap data dengan label yang diketahui, q_1 dan q_2 dapat diklasifikasikan (Cunningham & Delany, 2020).

Perhitungan jarak dengan tetangga terdekatnya dapat dituliskan dengan model matematis sebagai berikut:

$$d(q, x_i) = \sum_j w_j \delta(q_j, x_{if}) \quad (2.3)$$

$d(q, x_i)$ adalah perhitungan jarak dari data yang ingin diklasifikasikan q dan data yang telah diberikan label x . w_j adalah bobot klasifikasi, $\delta(q_j, x_{if})$ adalah penentuan

atribut klasifikasi antara q dan w (Cunningham & Delany, 2020).



Gambar 2.2.6. K-Nearest Neighbors untuk data banyaknya gaji bulanan tiap pekerja.

2.2.7 Loss Function

Loss function merupakan fungsi untuk menentukan seberapa besar kesalahan yang didapat dari *training* maupun *fitting* ML. Pada regresi, umumnya menggunakan *root mean square error* (RMSE) atau menggunakan *mean absolute error* (MAE). Penggunaannya berdasarkan model matematis yang ditunjukkan pada persamaan (2. 4) untuk RMSE dan persamaan (2. 5) untuk MAE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1} e_i^2} \quad (2. 4)$$

$$MAE = \frac{1}{n} \sum_{i=1} |e| \quad (2. 5)$$

RMSE dan MAE digunakan berdasarkan kebutuhan, dan jenis data yang digunakan. Sebagian merujuk bahwa MAE memiliki keunggulan untuk menunjukkan *loss function* yang stabil sehingga dapat merepresentasikan

data dengan benar, tetapi tidak sedikit pula yang masih menggunakan RMSE sebagai standar *loss function* (Chai & Draxler, 2014).

2.3 Python

Python merupakan bahasa pemrograman tingkat tinggi yang memiliki banyak *library* untuk mendukung penelitian. Bahasa tingkat tinggi berarti Python mudah dipahami oleh pengguna, karena cenderung lebih familier dengan bahasa manusia. Walaupun tergolong bahasa tingkat tinggi, Python merupakan salah satu bahasa yang efektif digunakan untuk melakukan penelitian numerik. Walaupun tidak se-efisien Fortran atau C++, Python memiliki kelebihan lain dengan banyaknya *library* yang mendukung banyak penelitian, seperti penggunaan NumPy, Pandas, TensorFlow, dan lain sebagainya (Dubois et al., 2007).

2.3.1 TensorFlow

TensorFlow adalah bagian dari *library* yang disediakan oleh python. TensorFlow banyak digunakan untuk melakukan penelitian terkait dengan ML, karena tersedia banyak *syntax* yang mempermudah penggunaan ML. TensorFlow dapat melakukan pekerjaan paralel untuk mengerjakan tugas komputasi pada algoritma ML ketika proses *training* dilaksanakan.

TensorFlow dapat meningkatkan kinerja dari sumber daya komputasi, karena kalkulasi yang dilakukan dapat juga memanfaatkan GPU yang telah di-*support*, walaupun penggunaan CPU secara konvensional masih bisa digunakan (Abadi, 2016).

2.3.2 Keras

Keras adalah *library* yang dimiliki oleh bahasa python. Keras dapat berjalan pada Theano ataupun TensorFlow. Keras mempermudah pengerjaan pembuatan model *deep learning* untuk pengembangan dan penelitian. Karena berjalan pada TensorFlow, Keras juga mampu memanfaatkan sumber daya komputasi GPU, tidak hanya CPU, menjadikan Keras *library* pilihan untuk menjalankan model ML (Brownlee, 2019).

2.3.3 Scikit-Learn

Scikit-Learn adalah alternatif *library* yang dapat digunakan di Python untuk keperluan penelitian dan pengembangan ML. Scikit-Learn memiliki banyak *toolkit* untuk mempermudah penelitian terkait ML, dengan banyak metode yang bisa dipilih. Pada kasus klasifikasi, Scikit-Learn juga menyediakan *syntax* untuk menyelesaikan masalah klasifikasi dengan metode *K-Nearest Neighborss* (Pedregosa et al., 2014).

BAB III

METODOLOGI

3.1 Waktu dan Tempat Pelaksanaan

Penelitian ini dilakukan pada bulan Maret 2021 sampai bulan Mei 2021 di Jalan Srigading Dalam No. 58K Kota Malang.

3.2 Alat dan Bahan

Alat yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Sebuah Laptop Asus dengan Spesifikasi Processor Intel i7 770HQ CPU @ 2.8 GHz, 4 core(s), 8 Logical Processor(s), RAM DDR4 16.00 GB, SSD 512 GB, NVIDIA Geforce GTX 1050.
2. Sebuah *Operating System* (OS) Windows 10 Pro
3. Beberapa *library* Python yaitu NumPy, Pandas, Matplotlib, TensorFlow, Keras, Seaborn dan Scikit-Learn.
4. NVIDIA-CUDA untuk membuka akses penggunaan GPU pada *library* TensorFlow.
5. IDE (*Integrated Development Environment*) berupa Spyder dengan versi instalasi Python 3.8.3.
6. Data cuaca tiap jam yang didapatkan dari laman penyedia data cuaca dengan alamat: <https://www.visualcrossing.com/weather/weather-data-services#/>

3.3 Tahapan Penelitian

Pada penelitian ini terdapat beberapa tahapan dengan

keterangan lengkap sebagai berikut:

1. Persiapan Komputasi
2. Pengolahan Data
3. Pembuatan *Model Neural Network* dan *K-Nearest Neighbors*
4. Pelatihan *Neural Network* dan *K-Nearest Neighbors*
5. Pengujian *Neural Network* dan *K-Nearest Neighbors*
6. Evaluasi dan Visualisasi ML

3.3.1 Persiapan Komputasi

Sebelum penelitian dimulai, berbagai perangkat lunak dipersiapkan dan diinstalasi terlebih dahulu. Pertama perangkat lunak IDE Spyder dengan instalasi Python 3.8.3. Perangkat lunak selanjutnya adalah instalasi NVIDIA-CUDA agar GPU pada perangkat keras bisa digunakan untuk menjalankan komputasi guna melancarkan dan mempercepat proses komputasi. Selanjutnya instalasi *library* Python dengan membuka *Windows Command Prompt* (CMD) dan mengetikkan perintah sebagai berikut:

1. Pip install NumPy
2. Pip install Pandas
3. Pip Install Matplotlib
4. Pip install TensorFlow
5. Pip install Keras
6. Pip install seaborn
7. Pip install scikit-learn

Setelah instalasi *library* Python selesai dilakukan, maka IDE Spyder dapat dijalankan dan pembuatan kode program dapat dimulai.

3.3.2 Pengolahan Data

Data yang telah didapatkan dari laman penyedia data cuaca kemudian dibersihkan terlebih dahulu. Data yang didapatkan disaring untuk melihat apakah ada yang parameternya tidak memiliki nilai (NaN). Apabila ada, maka data tersebut dihapus, karena akan mempengaruhi kualitas dari *train* model yang akan dijalankan.

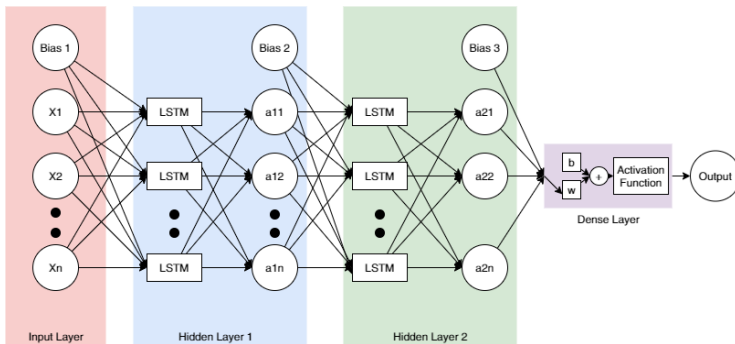
Selanjutnya dari data yang didapatkan diekspansi untuk memenuhi syarat *time-series neural network*, yaitu dengan mengelompokan data menjadi *time-step (lag)*. Misalkan *time-step* diatur selama 12 jam, maka akan ada 12 data uji kemudian data ke 13 menjadi label keluaran untuk dibandingkan hasilnya, lalu data bergeser dari 2-13 dengan data ke 14 menjadi label keluaran dan seterusnya. Pengolahan data ini bermaksud untuk memenuhi syarat *time series neural network*.

Setelah data tersebut selesai disiapkan, maka perlu diberikan skala agar mengurangi beban komputasi dan mengurangi *error*. Data dengan jangkauan (*range*) yang berbeda-beda diskalakan dengan ukuran 0-1, yang nantinya akan dikembalikan ke nilai sesungguhnya di akhir pengujian.

3.3.3 Pembuatan Model Neural Network dan K-Nearest Neighbors

Model NN yang dibuat memenuhi kriteria arsitektur regresi NN dengan metode LSTM. *Hyperparameter* seperti banyaknya unit dan layer, *loss function*, *optimizer*,

learning rate dan *activation function* akan disesuaikan tiap *training* dilaksanakan. Nilai awal yang di masukan sebelum penyesuaian adalah 1 *layer input*, 2 *hidden layer* LSTM, dan 1 *output layer*. Unit pada *hidden layer* pertama sebanyak 64, dan pada *hidden layer* kedua sebanyak 32. *optimizer* yang digunakan adalah ‘adam’ dengan opsi *learning rate* ‘default’. *Loss function* yang digunakan adalah RMSE dan MAE, kemudian dicari mana yang terbaik. *Activation function* yang digunakan adalah sigmoid dan relu, kemudian dicari mana yang terbaik. Gambaran arsitektur layer pada model NN dapat dilihat pada Gambar 3.3.1. Arsitektur LSTM dengan arah dari kiri ke kanan, dengan *input layer* berwarna biru, *hidden layer* dengan LSTM berwarna merah muda dan *output layer* berwarna hijau.



Gambar 3.3.1. Arsitektur LSTM

Model pada *K-Nearest Neighbors* dapat merujuk pada persamaan (2. 3). *K-Nearest Neighbors* tidak memiliki arsitektur khusus seperti *Neural Network* karena tidak memiliki neuron.

3.3.4 Pelatihan Neural Network dan K-Nearest Neighbors

Setelah model selesai dibuat dan data telah dibersihkan, maka proses pelatihan dapat dimulai. Pelatihan dimulai terlebih dahulu pada *Neural Network*, dengan membagi data *train* menjadi 60% dari data set dan 40% untuk data *test*. Apabila merujuk pada Gambar 2.2.5. Skema LSTM dan unit memorinya maka konstanta dari model akan didapatkan terlebih dahulu menggunakan data *train*. Tiap *epoch* atau pengulangan yang dilakukan, akan ada *update* bobot dan bias sesuai dengan persamaan (2. 5). Dengan menentukan jenis *loss function* RMSE (sebagai acuan awal), maka data akan diolah terus hingga nilai RMSE mengecil sembari memberikan *update* ke bias dan bobot. Apabila *epoch* telah selesai dilakukan, maka pelatihan *Neural Network* selesai.

Pelatihan *K-Nearest Neighbors* hanya dengan memberikan data yang telah diolah dan membaginya seperti *Neural Network*, lalu model akan pelatihan akan dilakukan dengan memindai daerah dan memberikan klasifikasi berdasarkan jarak ke tetangga terdekat. Setelah pelatihan selesai dengan 60% dari dataset, maka pelatihan selesai.

3.3.5 Pengujian Neural Network dan K-Nearest Neighbors

Pengujian dilakukan dengan menggunakan model yang telah di *train* dengan data set 60%, kemudian model tersebut digunakan untuk melakukan regresi pada *Neural Network*, dan klasifikasi pada *K-Nearest Neighbors* dengan menggunakan 40% data set yang telah dikhususkan untuk menjadi data uji. Hasil akhir yang

diharapkan adalah setelah *Neural Network* selesai menjalani pengujian, data regresi tersebut akan diklasifikasikan oleh *K-Nearest Neighbors*, kemudian akan dibandingkan dengan nilai sesungguhnya hingga tercipta prediksi cuaca.

3.3.6 Evaluasi dan Visualisasi ML

Setelah seluruh tahapan penelitian dilakukan, maka tahapan akhir adalah evaluasi dari model dan *hyperparameter* dari modelnya. Apabila hasil *loss function* masih terlalu tinggi, maka dapat dimodifikasi sesuai kebutuhan. Parameter tiap cuaca kemudian dibandingkan dan disesuaikan terhadap parameter memiliki korelasi paling tinggi antar parameter lainnya untuk mencapai prediksi cuaca yang optimal. Setelah selesai memberikan evaluasi dan menghasilkan hasil yang optimal, maka visualisasi dapat dilakukan.

Visualisasi dapat menggunakan *library* Matplotlib dan Seaborn untuk melihat perbandingan *loss function* pada data *training* dan data *test* sehingga dapat dievaluasi apakah model sudah *fit*, atau masih *undefit/overfit*. Selanjutnya visualisasi korelasi antara parameter fisis cuaca dibuat juga sebagai pembanding dari hasil prediksi tiap parameter yang digunakan.

BAB IV

JADWAL KEGIATAN PENELITIAN

4.1 Jadwal Kegiatan Penelitian

Berikut merupakan tabel susunan rencana kerja penelitian yang diajukan oleh penulis.

Tabel 4.1.1. Tabel Jadwal Kegiatan Penelitian

No	Kegiatan	Waktu											
		2021											
		3			4				5				
1	Pembuatan Proposal												
2	Persiapan Komputasi												
3	Pengolahan Data												
4	Pembuatan Model ML												
5	Pelatihan Model ML												
6	Pengujian Model ML												
7	Evaluasi dan Visualisasi ML												
8	Penulisan Tugas Akhir												

DAFTAR PUSTAKA

- Abadi, M. (2016). TensorFlow: learning functions at scale. *ACM SIGPLAN Notices*, 51(9), 1–1. <https://doi.org/10.1145/3022670.2976746>
- Brownlee, J. (2019, September 13). *Introduction to Python Deep Learning with Keras*. <https://machinelearningmastery.com/introduction-python-deep-learning-library-keras/>
- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. <https://doi.org/10.5194/gmd-7-1247-2014>
- Cogato, A., Meggio, F., Migliorati, M. D. A., & Marinello, F. (2019). Extreme weather events in agriculture: A systematic review. *Sustainability (Switzerland)*, 11(9), 1–18. <https://doi.org/10.3390/su11092547>
- Cunningham, P., & Delany, S. J. (2020). k-Nearest Neighbors classifiers 2nd edition (with python examples). *ArXiv*, 1, 1–22.
- Czum, J. M. (2020). Dive Into Deep Learning. *Journal of the American College of Radiology*, 17(5), 637–638. <https://doi.org/10.1016/j.jacr.2020.02.005>
- Dubois, P. F., Oliphant, T. E., Pérez, F., Granger, B. E., & Greenfield, P. (2007). *PYTHON: Guest Editor 's Introduction Python for Scientific Computing IPython: A System for Interactive Scientific Computing Reaching for the Stars with Python* (Issue June).
- Khosravi, A., Koury, R. N. N., Machado, L., & Pabon, J. J. G. (2018). Prediction of wind speed and wind direction using

- artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. *Sustainable Energy Technologies and Assessments*, 25(December 2017), 146–160.
<https://doi.org/10.1016/j.seta.2018.01.001>
- Kirono, D. G. C., Butler, J. R. A., McGregor, J. L., Ripaldi, A., Katzfey, J., & Nguyen, K. (2016). Historical and future seasonal rainfall variability in Nusa Tenggara Barat Province, Indonesia: Implications for the agriculture and water sectors. *Climate Risk Management*, 12, 45–58.
<https://doi.org/10.1016/j.crm.2015.12.002>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
<https://doi.org/10.1038/nature14539>
- Lions, J. L., Temam, R., & Wang, S. (1992). New formulations of the primitive equations of atmosphere and applications. *Nonlinearity*, 5(2), 237–288.
<https://doi.org/10.1088/0951-7715/5/2/001>
- Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning*. 1–38. <http://arxiv.org/abs/1506.00019>
- Lu, H., Li, Y., Chen, M., Kim, H., & Serikawa, S. (2017). *Brain Intelligence: Go Beyond Artificial Intelligence*. 24(2). <http://arxiv.org/abs/1706.01040>
- Morgan, P. (2018). Machine Learning Is Changing the Rules. In *O'Reilly* (Issue December).
<https://www.safaribooksonline.com/library/view/machine-learning-is/9781492035367/>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A.,

- Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2014). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 39(2014), i–ii.
- Potter, T. D., & Coleman, B. R. (2003). *HANDBOOK OF WEATHER, CLIMATE, AND WATER Dynamics, Climate, Physical Meteorology, Weather Systems, and Measurements*. John Wiley and Sons, Inc.
- Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning* (Second Edi). Packt Publishing Ltd. https://books.google.co.id/books?hl=en&lr=&id=_plGDwAAQBAJ&oi=fnd&pg=PP1&dq=python+machine+learning&ots=8tCNgWbmDH&sig=e4rQjE95vOAFhZeMIwTu6UdDIg&redir_esc=y#v=onepage&q&f=false
- Tosepu, R., Gunawan, J., Effendy, D. S., Ahmad, L. O. A. I., Lestari, H., Bahar, H., & Asfian, P. (2020). Correlation between weather and Covid-19 pandemic in Jakarta, Indonesia. *Science of the Total Environment*, 725, 138436. <https://doi.org/10.1016/j.scitotenv.2020.138436>
- Volokitin, A., Timofte, R., & Van Gool, L. (2016). Deep Features or Not: Temperature and Time Prediction in Outdoor Scenes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1136–1144. <https://doi.org/10.1109/CVPRW.2016.145>
- Watts, A. (2014). *The weather handbook* (3rd ed.). Adlard Coles Nautical Press. https://books.google.co.id/books?id=c63vAwAAQBAJ&dq=The Weather Handbook&source=gbs_book_other_versions
- Wirjohamidjojo, S., & Swarinoto, Y. (2010). *Iklim Kawasan Indonesia (Dari Aspek Dinamik - Sinoptik)*. Badan Meteorologi Klimatologi dan Geofisika.

Zhang, Z., Ma, H., Fu, H., & Zhang, C. (2016). Scene-free multi-class weather classification on single images. *Neurocomputing*, 207, 365–373. <https://doi.org/10.1016/j.neucom.2016.05.015>