KAPITA SELEKTA

Prediksi Penyebaran Penyakit Demam Berdarah pada Data *DengAI* dengan *Machine Learning* pada Mata Kuliah Kapita Selekta

Ditulis untuk memenuhi sebagian persyaratan akademik guna
menyelesaikan mata kuliah Kapita Selekta

Oleh:

Osvaldo Figo	01112180010
Terry Hilario Santoso	01112180028
Yudiestira Dwi Sentosa	01112180030

PROGRAM STUDI MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS PELITA HARAPAN

TANGERANG

2021

i. *Executive Summary*

Dalam eksperimen kali ini, telah dilakukan uji atau percobaan untuk membuat model yang digunakan dalam memprediksi jumlah total kasus demam berdarah di kota San Juan dan Iquitos di Negara Peru. Demam berdarah adalah salah satu penyakit yang menular melalui perantara nyamuk. Penyakit ini memiliki tingkat risiko kematian yang cukup besar dan menyebar cukup cepat. Dikarenakan penyakit demam berdarah diperantarai oleh nyamuk, maka terdapat korelasi terhadap faktor geografis dan cuaca terhadap laju perkembangan penyakit ini. Lokasi geografis yang berada di sekitar khatulistiwa dimana di lokasi tersebut curah hujan cukup tinggi, membuat nyamuk sangat mudah untuk berkembang biak. Dari data yang sudah dikumpulkan, hampir 500 juta kasus demam berdarah setiap tahunnya timbul di daerah Amerika Latin. Angka tersebut cukup besar dan dengan banyaknya kelompok masyarakat yang tertular dengan penyakit demam berdarah membuat fasilitas kesehatan tidak dapat menampung semua pasien karena kapasitas kamar yang telah terisi penuh. Dengan alasan yang sudah dipaparkan sebelumnya, menjadi dasar dari eksperimen ini. Diharapkan dengan adanya model prediksi, pemerintah dan seluruh lapisan masyarakat dapat bersia[untuk menghadapi penyakit ini.

Model prediksi menggunakan metode *random forest* atau lebih tepatnya menggunakan fungsi *RandomForestRegressor*. Data yang diambil dari *drivendata.org* sehingga data dianggap valid. Dalam proses *data preprocessing* dilakukan pengisian data yang kosong, pemilihan kolom data yang diperlukan, dan perubahan jenis data teks ke data numerik. Model data didapatkan dari data *train* dengan rasio 80% data *test* dan 20% data *train*. Kemudian dari model data yang sudah didapatkan, dimasukkan data *test* yang sesungguhnya dan didapatkanlah hasil akhir. Hasil yang sudah didapatkan kemudian diverifikasi dan mendapatkan MAE 26,09.

DAFTAR ISI

i. ***Executive Summary*** 1

I. Pendahuluan 3

II. Metodologi Penelitian 4

2.1. Pengumpulan dan Pembagian Data 5

2.2. ***Random Forest*** 6

2.3. Pengujian dan Evaluasi Model 8

2.4. Pengujian ke Data Penguji Aktual 8

III. Analisis dan Pembahasan 9

3.1. Data 9

3.2. ***Features*** **dalam dataset** 9

3.3. ***Data Preprocessing*** 11

3.4. Aplikasi Model ***Random Forest*** 18

IV. Penutup 20

4.1. Kesimpulan 20

4.2 Saran 20

DAFTAR PUSTAKA 22

Lampiran 1 – Kode 23

Lampiran 2 – Hasil Data Prediksi 29

Lampiran 3 - Nilai kesamaan penulisan dari Turnitin. 35

I. Pendahuluan

Penyakit demam berdarah adalah penyakit yang berasal dari nyamuk dimana penyakit ini terjadi pada negara beriklim tropis dan sub tropis. Dalam kasus yang biasa terjadi, gejala demam berdarah mirip dengan gejala flu diantara lain seperti panas, ruam, dan nyeri sendi dan otot. Pada kasus yang parah, penyakit demam berdarah dapat menyebabkan pendarahan akut, tekanan darah rendah dan kematian. Dikarenakan penyakit ini dibawa oleh nyamuk, kecepatan transmisi penyebaran penyakit demam berdarah tergantung dengan variabel cuaca seperti temperatur dan curah hujan. Meskipun hubungan dari cuaca sangat kompleks, tetapi semakin banyak peneliti yang berargumen jika perubahan cuaca dapat mempengaruhi laju penyebaran dari penyakit demam berdarah. Dengan pengetahuan akan cuaca dan distribusi penyebaran penyakit diharapkan memberikan dampak yang signifikan dari fasilitas kesehatan.

Dalam beberapa tahun terakhir, penyakit demam berdarah sudah menyebar ke banyak negara. Secara historis, penyakit tersebut umumnya menyerang wilayah Asia Tenggara dan Kepulauan di Samudera Pasifik. Pada saat ini, hampir setengah miliar kasus per tahun terjadi di Amerika Latin. Dengan menggunakan data yang diambil dari berbagai lembaga Amerika *U.S. Federal Government agencies—from the Centers for Disease Control and Prevention to the National Oceanic and Atmospheric Administration* yang merupakan bagian dari *U.S. Department of Commerce*, akan dilakukan prediksi jumlah kasus demam berdarah yang terjadi setiap minggunya di kota San Juan dan kota Iquitos di Peru.

Diharapkan juga nantinya dengan pemahaman hubungan dari cuaca dan penyakit demam berdarah dapat meningkatkan inisiatif untuk melakukan penelitian pada variabel yang berpengaruh dan membantu perencanaan alokasi biaya untuk menanggulangi pandemi ini.

II. Metodologi Penelitian

Pada bagian ini akan dijelaskan mengenai langkah-langkah yang akan dilakukan guna memprediksi penyebaran demam berdarah dengan menggunakan metode *random forest*. Diagram 2.1 merupakan diagram yang menjelaskan proses dalam melakukan prediksi.

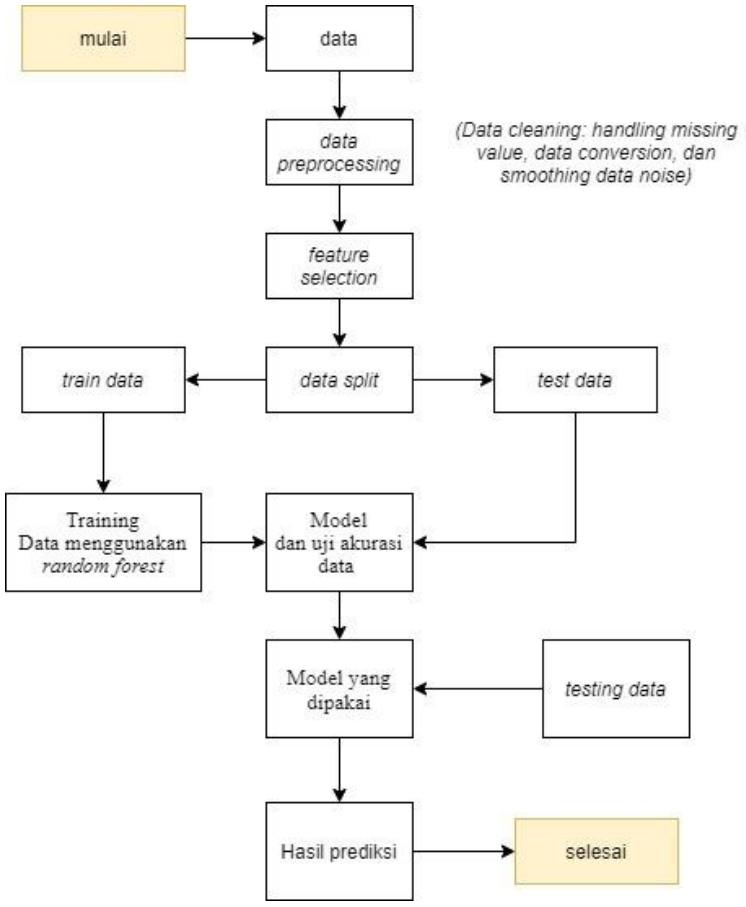


Diagram 2.1: *flowchart* pengolahan data

2.1. Pengumpulan dan Pembagian Data

Penelitian ini menggunakan *dataset* yang berasal dari situs *drivendata.org* lebih tepatnya diambil kompetisi *DengAI: Predicting Disease Spread*. Data yang diberikan terdapat empat buah data yaitu *features train*, *features label*, *feature test*, dan contoh format untuk pengumpulan. Langkah pertama yang dilakukan adalah menyatukan data *features train* dan *features label*, kemudian akan dilakukan analisis data yang masuk ke dalam langkah *preproccesing*. Akan dilakukan beberapa hal penting yang dilakukan dalam langkah ini antara lain:

- *handling missing values*, langkah ini bertujuan untuk menentukan apa aksi yang harus dilakukan untuk mengatasi data-data yang kosong. Penelitian ini mengisi data-data yang kosong tersebut menggunakan *mean* atau median dari kelas (kolom) yang sama dengan melihat distribusi dari *features* tersebut,
- *smoothing noisy data*, langkah ini menganalisis data-data yang merupakan pencilan dan menentukan apakah data-data tersebut harus dibuang atau tetap dipakai karena pertimbangannya adalah bisa aja memang data yang abnormal,
- *data conversion*, di mana dalam langkah ini akan dilakukan pengubahan bentuk data ke bentuk numerik, karena model *machine learning* hanya dapat mengolah data yang numerik. Dalam kasus penelitian ini, akan diubah data-data dalam kolom "*city*" yaitu "sj" dan "iq" menjadi 0 dan 1.

Selanjutnya, data *train* yang sudah dibersihkan akan dibagi menjadi dua yaitu data pelatihan (*data train*) dan data penguji (*data test*) dengan proporsi 80% sebagai data pelatihan dan 20% data penguji. Data-data ini akan dipilih secara acak dan seluruh proses akan dilakukan dengan program Python.

2.2. *Random Forest*

Algoritma pembelajaran yang dipilih dalam memodelkan prediksi data adalah model *random forest*. Dalam program Python akan digunakan *package sklearn.ensemble* dan menggunakan kernel *RandomForestRegressor*. Penelitian ini menggunakan model *random forest* yang *default* tanpa mengubah parameter.

Bentuk default dari *random forest* di Python adalah jumlah pohon yang akan dibuat dalam penelitian ini ada total 100 pohon atau yang biasa dilambangkan dengan notasi B, kemudian jumlah kedalaman pohon yang akan dibuat pada setiap pohon akan bervariasi karena pengaturan *default* dibuat *none* yang berarti akan dibuat kedalaman maksimum hingga semua *leaves* sudah lebih kecil dari *minimum samples split* di mana dalam kasus ini karena bentuk data adalah *integer* maka nilai *minimum samples split* akan dipakai angka yang paling kecil. Karena model *random forest* merupakan sebuah gabungan dari pohon-pohon, maka *base estimator* dari model *random forest* adalah tidak lain *decision tree*.

Algoritma dari random forest kumpulan data berbasis pohon yang dimana setiap pohonnya tergantung oleh variabel acak. Untuk lebih jelas buat setiap n-dimensi random vektor $X = (X1, \ldots, Xn)^T$ mewakili variabel input atau prediktor bernilai riil dan variabel acak Y yang mewakili respons yang dihargai nyata, dengan mengasumsikan *joint distribution* yang tidak diketahui $P_{XY}(X, Y)$. Tujuannya adalah untuk menemukan fungsi prediksi (X) untuk memprediksi Y. Fungsi prediksi ditentukan oleh *loss function* $L(Y, f(X)))$ dan didefinisikan untuk meminimalkan nilai kerugian yang diharapkan

$$E_{XY}(L(Y, f(X))) \tag{2.1}$$

L adalah loss squared error $L(Y, f(X)) = \left(Y - f(X)\right)^2$ untuk regresi dan kerugian nol-satu untuk klasifikasi

$$L\big(Y, f(X)\big) = I\big(Y \neq f(X)\big) = \begin{cases} 0 \; if \; Y = f(x) \\ 1 \; otherwise \end{cases} \tag{2.2}$$

Ternyata pada meminimalkan E_{XY} $(L(Y, f(X)))$untuk *squared error loss* memberikan ekspektasi bersyarat

$$f(x) = E(Y|X = x) \tag{2.3}$$

atau dikenal sebagai fungsi regresi. Dalam situasi klasifikasi, jika kumpulan nilai Y yang mungkin ditandai dengan Y_, meminimalkan $E_{XY}(L\big(Y, f(X)\big))$ untuk nol-satu kerugian memberikan

$$f(x) = \arg\max_{y \in Y\_} P(Y = y|X = x) \tag{2.4}$$

dikenal sebagai *Bayes rule*

untuk membangun f dalam hal koleksi yang disebut “Base Learner" h1(x),...,hn(x) dan Base learner ini digabungkan untuk memberikan “ensemble pre-dictor” f(x). Dalam regresi, base learner rata-rata

$$f(x) = \frac{1}{J}\sum_{J=1}^{J} h_j(x) \tag{2.5}$$

2.3. Pengujian dan Evaluasi Model

Model yang sudah berhasil didapatkan dari hasil *training data* kemudian akan diuji dengan data penguji. Hasil prediksi dari model akan dicatat dan nantinya akan dibandingkan keakurtannya dengan hasil yang sebenarnya, kemudian akan dihitung nilai R-squared

$$R^2 = 1 - \frac{RSS}{TSS}, \tag{2.6}$$

dan MAE (*mean absolute error*)

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}, \tag{2.7}$$

dari kedua data (data pelatihan dan data penguji).

2.4. Pengujian ke Data Penguji Aktual

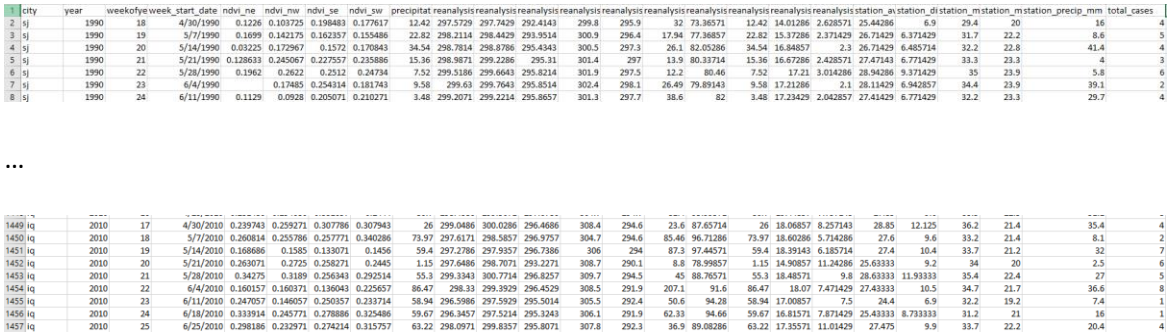
Model yang sudah didapatkan kemudian dipakai untuk menguji *data test* dari *dengue_features_test.csv* untuk kemudian dikumpulkan ke situs *drivendata.org*. Data yang sudah berhasil diprediksi kemudian akan diubah ke bentuk yang sesuai dengan contoh *submission* yang sudah diberikan berisi kolom "*city*", "*year*", "*week of year*", dan hasil prediksi di kolom "*total_cases*". Data akhir yang dikumpulkan akan kembali diuji menggunakan MAE sebagai perhitungan skor dan penempatan peringkat pada kompetisi *DengAI: Predicting Disease Spread*.

III. Analisis dan Pembahasan

Pada bab ini dijelaskan hasil dari implementasi data terhadap metode *Random Forest* serta analisisnya. Dimulai dengan penjelasan data-data yang akan digunakan pada penelitian ini , dilanjutkan dengan hasil evaluasi dari metode *Random Forest.*

3.1. Data

Pada penelitian ini digunakan empat ratus tujuh belas data yang diambil dari situs *drivendata.org* untuk mengikuti kompetisi ini. Data yang digunakan adalah data yang diambil di dua kota yaitu kota San Juan dan Iquitos yang dapat digunakan untuk memprediksi tingkat penyebaran penyakit demam berdarah. Tampilan data yang akan digunakan ada pada Gambar 3.1.



	city	year	weekofye	week_start_date	ndvi_ne	ndvi_nw	ndvi_se	ndvi_sw	precipitat	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	station_a	station_di	station_m	station_m	station_precip_mm	total_cases
2	sj	1990	18	4/30/1990	0.1226	0.103725	0.198483	0.177617	12.42	297.5729	297.7429	292.4143	299.8	295.9	32	73.36571	12.42	14.01286	2.628571	25.44286	6.9	29.4	20	16	4
3	sj	1990	19	5/7/1990	0.1699	0.142175	0.162357	0.155486	22.82	298.2114	298.4429	293.9514	300.9	296.4	17.94	77.36857	22.82	15.37286	2.371429	26.71429	6.371429	31.7	22.2	8.6	5
4	sj	1990	20	5/14/1990	0.03225	0.172967	0.1572	0.170843	34.54	298.7814	298.8786	295.4343	300.5	297.3	26.1	82.05286	34.54	16.84857	2.3	26.71429	6.485714	32.2	22.8	41.4	4
5	sj	1990	21	5/21/1990	0.128633	0.245067	0.227557	0.235886	15.36	298.9871	299.2286	295.31	301.4	297	13.9	80.33714	15.36	16.67286	2.428571	27.47143	6.771429	33.3	23.3	4	3
6	sj	1990	22	5/28/1990	0.1962	0.2622	0.2512	0.24734	7.52	299.5186	299.6643	295.8214	301.9	297.5	12.2	80.46	7.52	17.21	3.014286	28.94286	9.371429	35	23.9	5.8	6
7	sj	1990	23	6/4/1990		0.17485	0.254314	0.181743	9.58	299.63	299.7643	295.8514	302.4	298.1	26.49	79.89143	9.58	17.21286	2.1	28.11429	6.942857	34.4	23.9	39.1	2
8	sj	1990	24	6/11/1990	0.1129	0.0928	0.205071	0.210271	3.48	299.2071	299.2214	295.8657	301.3	297.7	38.6	82	3.48	17.23429	2.042857	27.41429	6.771429	32.2	23.3	29.7	4

...

1449	iq	2010	17	4/30/2010	0.239743	0.259271	0.307786	0.307943	26	299.0486	300.0286	296.4686	308.4	294.6	23.6	87.65714	26	18.06857	8.257143	28.85	12.125	36.2	21.4	35.4	4
1450	iq	2010	18	5/7/2010	0.260814	0.255786	0.257771	0.340286	73.97	297.6171	298.5857	296.9757	304.7	294.6	85.46	96.71286	73.97	18.60286	5.714286	27.6	9.6	33.2	21.4	8.1	2
1451	iq	2010	19	5/14/2010	0.168686	0.1585	0.133071	0.1456	59.4	297.2786	297.9357	296.7386	306	294	87.3	97.44571	59.4	18.39143	6.185714	27.4	10.4	33.7	21.2	32	7
1452	iq	2010	20	5/21/2010	0.263071	0.2725	0.258271	0.2445	1.15	297.6486	298.7071	293.2271	308.7	290.1	8.8	78.99857	1.15	14.90857	11.24286	25.63333	9.2	34	20	2.5	6
1453	iq	2010	21	5/28/2010	0.34275	0.3189	0.256343	0.292514	55.3	299.3343	300.7714	296.8257	309.7	294.5	45	88.76571	55.3	18.48571	9.8	28.63333	11.93333	35.4	22.4	27	5
1454	iq	2010	22	6/4/2010	0.160157	0.160371	0.136043	0.225657	86.47	298.33	299.3929	296.4529	308.5	291.9	207.1	91.6	86.47	18.07	7.471429	27.43333	10.5	34.7	21.7	36.6	8
1455	iq	2010	23	6/11/2010	0.247057	0.146057	0.250357	0.233714	58.94	296.5986	297.5929	295.5014	305.5	292.4	50.6	94.28	58.94	17.00857	7.5	24.4	6.9	32.2	19.2	7.4	1
1456	iq	2010	24	6/18/2010	0.333914	0.245771	0.278886	0.325486	59.67	296.3457	297.5214	295.3243	306.1	291.9	62.33	94.66	59.67	16.81571	7.871429	25.43333	8.733333	31.2	21	16	1
1457	iq	2010	25	6/25/2010	0.298186	0.232971	0.274214	0.315757	63.22	298.0971	299.8357	295.8071	307.8	292.3	36.9	89.08286	63.22	17.35571	11.01429	27.475	9.9	33.7	22.2	20.4	4

Gambar 3.1 *Data features dengue* San Juan dan Iquitos

3.2. *Features* dalam dataset

Dalam dataset terdapat beberapa *features* yang sudah disediakan. Setiap *features* atau kolom, akan dijelaskan sebagai berikut.

- city – terdapat singkatan dalam data: sj untuk San Juan dan iq untuk Iquitos.
- week_start_date – tanggal yang diberikan dalam format *yyyy-mm-dd.*

Berikut beberapa *features* yang datanya diambil dari satelit NOAA's GHCN yang berfungsi untuk mengukur data cuaca secara harian pada kota terkait.

- station_max_temp_c – temperatur maksimum dalam derajat celcius.
- station_min_temp_c – temperatur minimum dalam derajat celcius.
- station_avg_temp_c – rata-rata temperatur dalam derajat celsius.
- station_precip_mm – total curah hujan dalam milimeter.
- station_diur_temp_rng_c – rentang temperatur haruan dalam derajat celcius.

Berikut *features* yang datanya diambil dari satelit PERSIANN yang berfungsi untuk mengukur curah hujan secara harian (0.25x0.25 skala derajat).

- precipitation_amt_mm – total curah hujan dalam milimeter

Berikut *features* yang datanya diambil dari satelit NOAA's NCEP *Climate Forecast System Reanalysis measurements* (0.5x0.5 skala derajat).

- reanalysis_sat_precip_amt_mm – Curah hujan total dalam milimeter
- reanalysis_dew_point_temp_k – Rata-rata suhu titik embun dalam kelvin
- reanalysis_air_temp_k – Rata-rata suhu udara dalam kelvin
- reanalysis_relative_humidity_percent – Persentase rata-rata kelembaban relatif
- reanalysis_specific_humidity_g_per_kg – Rata-rata kelembaban spesifik
- reanalysis_precip_amt_kg_per_m2 – Curah hujan total
- reanalysis_max_air_temp_k – Suhu udara maksimum
- reanalysis_min_air_temp_k – Suhu udara minimum

- reanalysis_avg_temp_k – Suhu udara rata-rata
- reanalysis_tdtr_k – Kisaran suhu harian

Berikut *features* yang datanya diambil dari satelit Normalized difference vegetation index (NDVI) yang berfungsi untuk mengukur Indeks Vegetasi Perbedaan Normal CDR NOAA's (skala 0,5x0,5 derajat).

- ndvi_se – Piksel tenggara centroid kota
- ndvi_sw – Piksel barat daya centroid kota
- ndvi_ne – Piksel timur laut centroid kota
- ndvi_nw – Piksel barat laut kota sentroid

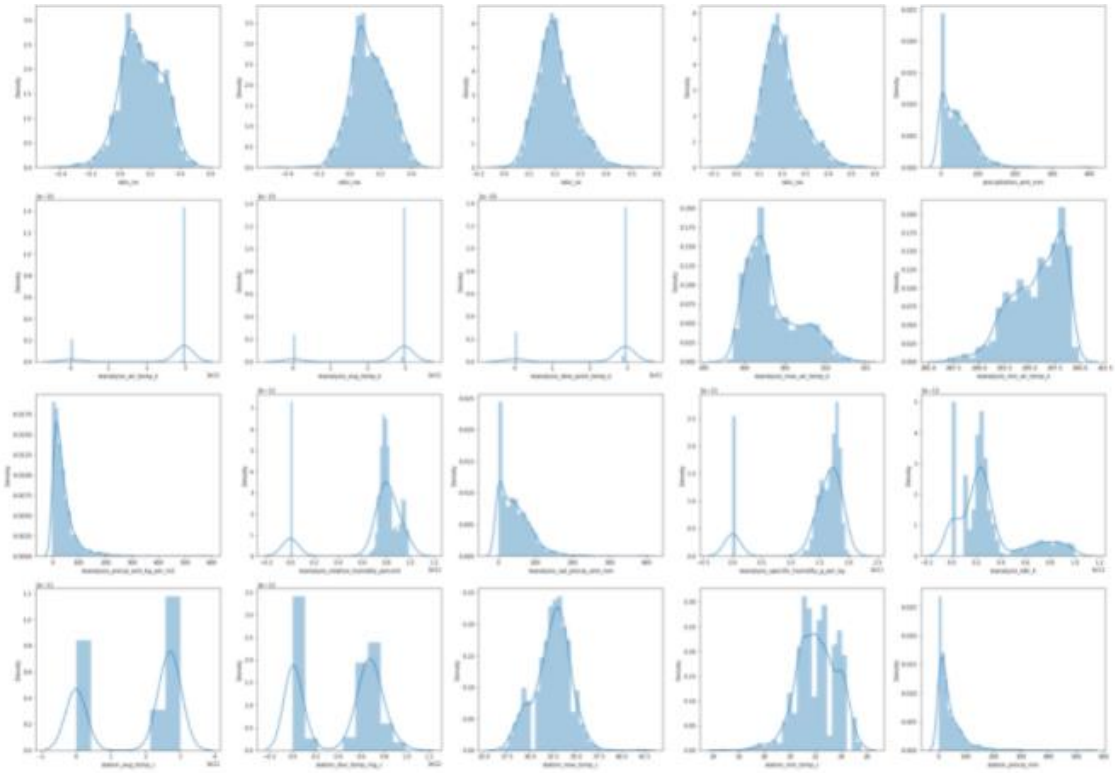
3.3. *Data Preprocessing*

Melakukan *data preprocessing* memiliki peran penting dalam pengolahan data. Perlu adanya pengecekan terhadap kualitas data sehingga proses dari pengolahan data tidak terganggu dan dapat menghasilkan hasil yang optimal. Terdapat beberapa teknik *data preprocessing*, diantaranya *Data Cleaning, Data Integration, Data Reduction*, *Data Transformation* dan *Data Discretization*. Dalam data yang dimiliki, teknik *data processing* yang digunakan adalah *Data Cleaning* dan *Data Transformation*.

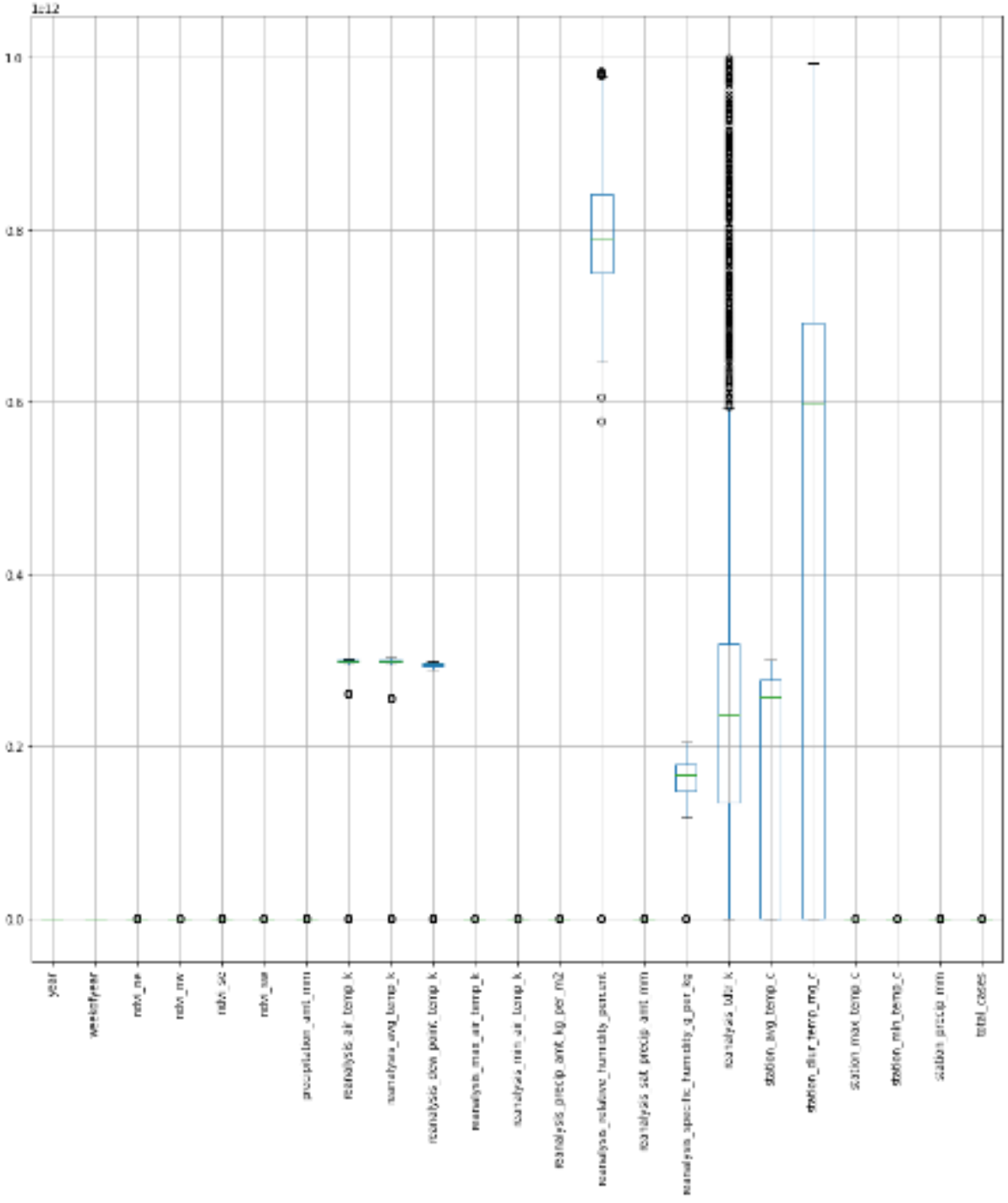
Proses *data cleaning* dimulai dengan mengecek kolom yang memliki data kosong atau kurang lengkap. Dalam kasus ini data dianggap valid sehingga tidak ada pengurangan data yang perlu dilakukan. Dari hasil pengecekan pada Gambar 3.2, terdapat dua puluh *features* (x) yang kosong, oleh karena itu akan dilakukan *handling missing values* dengan melihat bentuk distribusi tiap *features* pada Gambar 3.3. Kemudian setiap baris yang kosong akan diisi dengan perintah df.fillna.(filler). Namun, sebelum mengisi data yang kosong akan dilakukan pemilihan features terlebih dahulu dengan uji korelasi.

city	0
year	0
weekofyear	0
week_start_date	0
ndvi_ne	194
ndvi_nw	52
ndvi_se	22
ndvi_sw	22
precipitation_amt_mm	13
reanalysis_air_temp_k	10
reanalysis_avg_temp_k	10
reanalysis_dew_point_temp_k	10
reanalysis_max_air_temp_k	10
reanalysis_min_air_temp_k	10
reanalysis_precip_amt_kg_per_m2	10
reanalysis_relative_humidity_percent	10
reanalysis_sat_precip_amt_mm	13
reanalysis_specific_humidity_g_per_kg	10
reanalysis_tdtr_k	10
station_avg_temp_c	43
station_diur_temp_rng_c	43
station_max_temp_c	20
station_min_temp_c	14
station_precip_mm	22
dtype: int64	

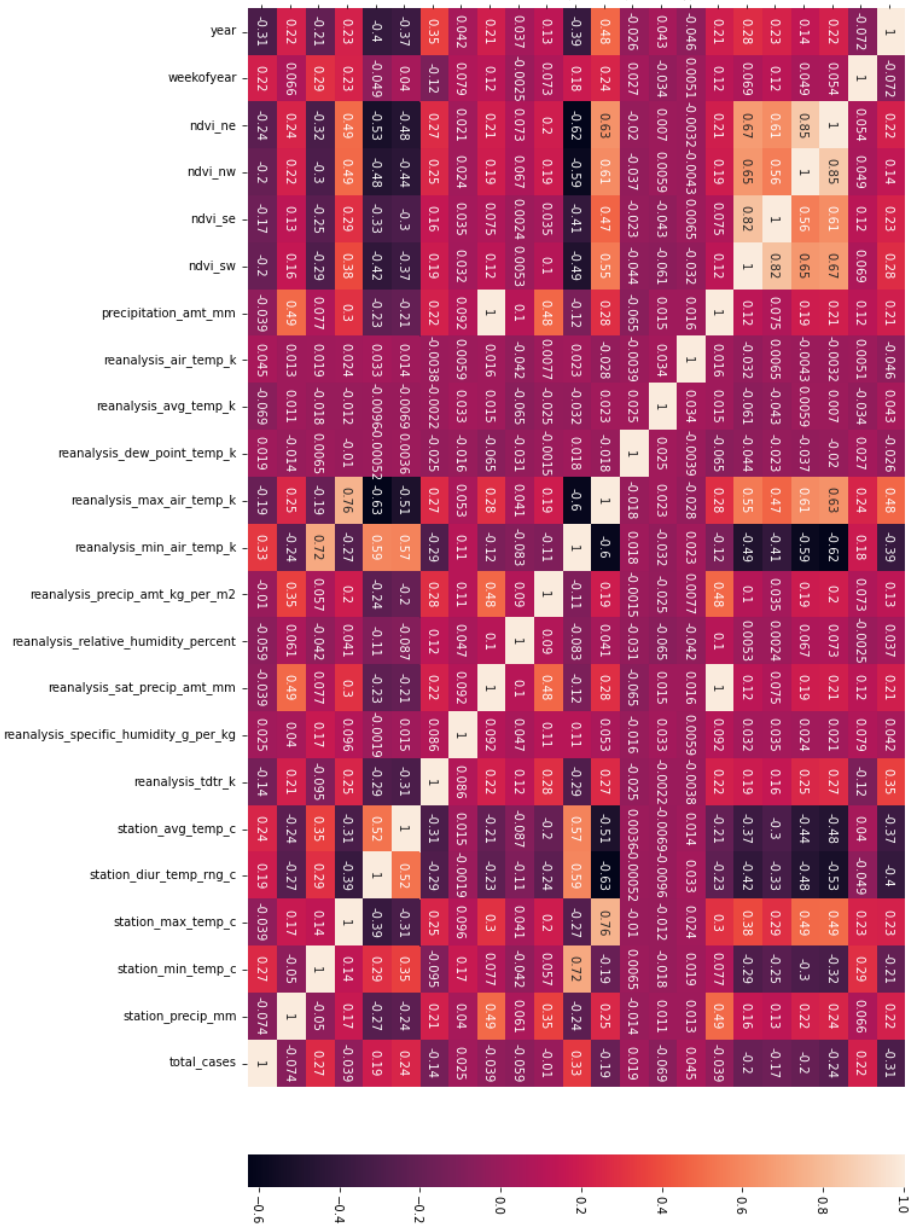
Gambar 3.2 hasil pengecekan kolom kosong data *train*



Gambar 3.3 ditribusi data tiap *features*



Gambar 3.4 *boxplot* tiap *features*

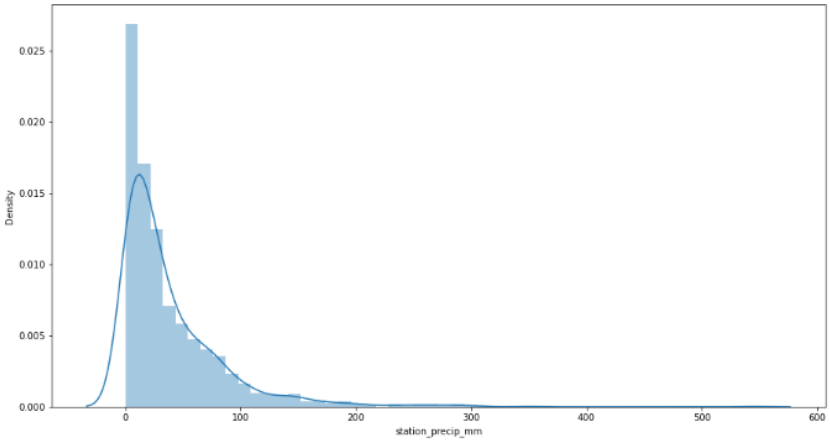

year
weekofyear
ndvi_ne
ndvi_nw
ndvi_se
ndvi_sw
precipitation_amt_mm
reanalysis_air_temp_k
reanalysis_avg_temp_k
reanalysis_dew_point_temp_k
reanalysis_max_air_temp_k
reanalysis_min_air_temp_k
reanalysis_precip_amt_kg_per_m2
reanalysis_relative_humidity_percent
reanalysis_sat_precip_amt_mm
reanalysis_specific_humidity_g_per_kg
reanalysis_tdtr_k
station_avg_temp_c
station_diur_temp_rng_c
station_max_temp_c
station_min_temp_c
station_precip_mm
total_cases
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Gambar 3.5 *heat map* korelasi antar *features*

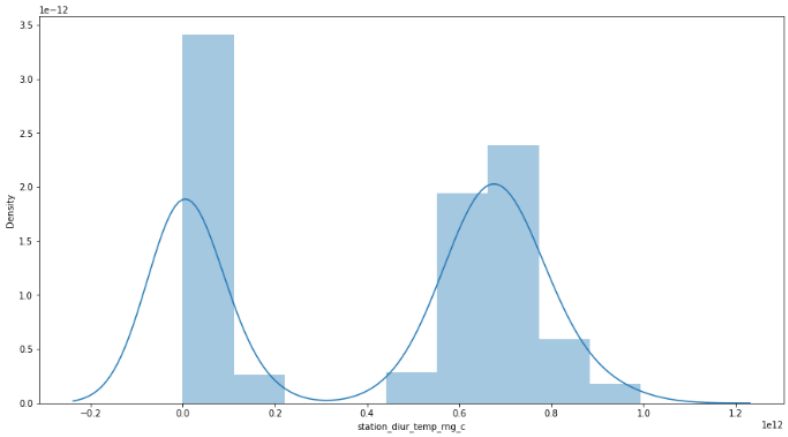
Hasil dari *boxplot* pada Gambar 3.4 menunjukkan bahwa terdapat pencilan dalam *features*, namun hasil ini akan dibaikan dengan asumsi masih dalam batas wajar. Kemudian, akan dicari korelasi antara variabel respon dengan seluruh variabel predictor dan akan ditampilkan *heatmap* pada Gambar 3.5 yang akan langsung menjelaskan korelasi antar variabel. Setelah menampilkan *heatmap* akan dipilih beberapa variabel prediktor yang mempunyai korelasi tinggi dengan variabel respon. Kemudian, variabel dengan korelasi yang tinggi dipilih dan dimasukkan ke dalam *array*. Terdapat tujuh *features* yang terpilih untuk terpakai dalam membuat model yang akan digunakan untuk prediksi hasil dari *dengue_features_test*. Tujuh *feature* yang terpiliih di antara lain:

1. *city*,
2. *station_precip_mm*,
3. *station_diur_temp_rng_c*,
4. *station_avg_temp_c*,
5. *reanalysis_min_air_temp_k*,
6. *year*, dan
7. *Weekofyear*.

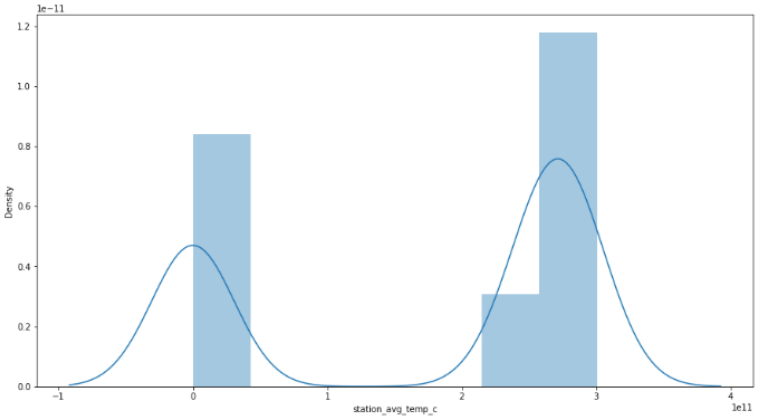
Kemudian akan ditinjau bentuk distribusi dari tiap *features* yang terpilih untuk mengisi data yang kosong. Hasil yang diperoleh yakni distribusi normal untuk *features station_precip_mm* dan *reanalysis_min_air_temp_k* serta distribusi lainnya untuk *features station_diur_temp_rng_c* dan *station_avg_temp_c*. *Features* dengan persebaran data distribusi normal, pada data yang kosong akan diisi menggunakan nilai rata-rata *features* dan distribusi lainnya akan menggunakan nilai median *features*.



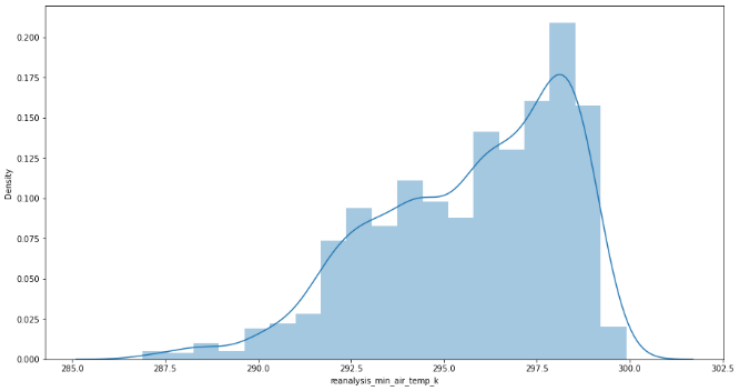
Gambar 3.6: Distribusi variabel *station_precip_mm*



Gambar 3.7: Distribusi variabel s*tation_diur_temp_rng_c*



Gambar 3.8: Distribusi variabel *station_avg_temp_c*



Gambar 3.9: Distribusi variabel *reanalysis_min_air_temp_k*

Setelah data sudah bersih dan tidak terdapat lagi kolom yang kosong, sebenarnya data sudah dapat diproses, namum terdapat data yang merupakan tipe string, yang dimana data ini tidak dapat digunakan dalam fungsi *RandomForestRegressor* yang hanya menerima tipe data angka. Maka dilakukan *data transformation*, yang akan mengkonversi data pada kolom "*city*" menjadi data numerik pada Gambar 3.10 sehingga data dapat digunakan dalam fungsi *RandomForestRegressor*.

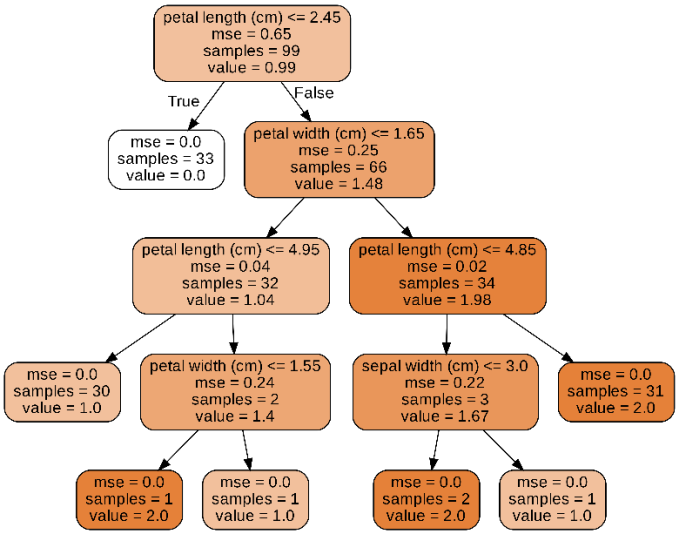
	city
0	0
1	0
2	0
3	0
4	0
...	...
1451	1
1452	1
1453	1
1454	1
1455	1

	city
0	sj
1	sj
2	sj
3	sj
4	sj
...	...
1451	iq
1452	iq
1453	iq
1454	iq
1455	iq

Gambar 3.10 mengubah data kolom "*city*"

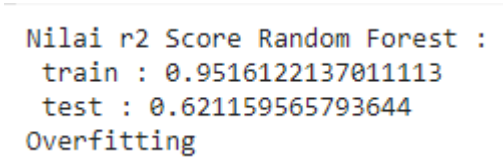
3.4. Aplikasi Model *Random Forest*

Data yang sudah selesai dibersihkan kemudian akan dipisah menjadi dua yaitu data pelatihan dan data pengujian dengan proporsi 8:2. Kemudian data *train* ini akan dipakai untuk membuat model *random forest regressor*. Model dari *random forest* akan berbentuk pohon *decision tree* dan akan dibuat sejumlah seratus pohon, dengan kedalaman pohon semaksimal mungkin sampai data sudah tidak dapat di*split*. Salah satu model pohon yang dihasilkan dapat dilihat pada Gambar 3.11, model ini hanya sebagai perwakilan model pohon lain.



Gambar 3.11: Bentuk model *decision tree* yang dihasilkan

Model *random forest* ini mendapatkan hasil R^2 yang dapat dilihat pada Gambar 3.12. Meskipun model dinyatakan *overfitting* oleh peneliti karena margin *train* dan *test* melebihi 0.03, namun model mendapatkan hasil $MAE = 26.0986$.

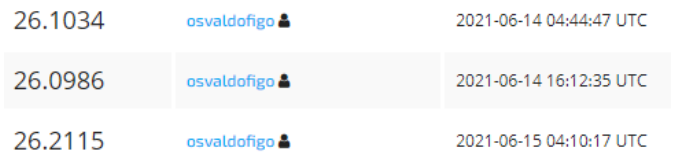


Gambar 3.12: Hasil perhitungan R^2

IV. Penutup

4.1. Kesimpulan

Penelitian ini telah melakukan pemodelan dengan metode *random forest* untuk memprediksi penyebaran penyakit demam berdarah, namun model yang dihasilkan masih dinilai sebagai model yang *overfit* dengan perbangingan hasil R^2 kurang lebih 95% untuk *data train* split dan 60% untuk *data test* split sehingga menghasilkan margin kurang lebih 35% serta perhittungan nilai R^2 yang relatif rendah untuk hasil *data test split*. Hasil MAE yang didapat dari model *random forest* cenderung berada di angka 26 setelah dilakukan tiga kali percobaan.



Gambar 4.1: Hasil MAE pada situs *drivendata.org*

4.2 Saran

Metode yang digunakan untuk membersihkan data dan memodelkan algoritama dalam penelitian ini seharusnya masih dapat dikembangkan agar dapat memberikan hasil prediksi yang lebih akurat serta menghilangkan kemungkinan-kemungkinan terjadinya *overfitting* dan *underfitting*. Beberapa hal yang dapat dilakukan untuk mengembangkan penelitian ini adalah sebagai berikut:

1. menggunakan metode PCA untuk memanipulasi *features* sehingga dapat mereduksi jumlah kolom tanpa menghilangkan atau dengan tetap mempertahankan infomarsi yang ada,
2. menganalisis data-data pencilan secara mendalam agar mendapatkan model yang lebih baik,

3. menggunakan metode lain seperti *probit regression*, *support vector machine* (SVM), naïve bayes, k-*nearest-neighbors*, dan sebagainya.

DAFTAR PUSTAKA

Prasetyo, Hendro. Juni 2019. Artikel : *Apa Itu Preprocessing*. Diakses pada tanggal 18 Juni 2021. https://hendroprasetyo.com/apa-itu-preprocessing/#.YMyyiWgzY2w

Zhang, C. and Ma, Y., 2011. *Random Forests*. ResearchGate. Diakses pada tanggal 18 Juni 2021. https://www.researchgate.net/publication/236952762_Random_Forests

Lampiran 1 – Kode

Data Cleaning

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import scipy.stats as ss

%matplotlib inline

df = pd.read_excel("dengue_features_train.xlsx")
pd.options.display.max_columns = 50
df.head()

df.isna().sum()

fig = plt.figure(figsize=(35,25))
kolom = df.columns[4:]
for i in range(len(kolom)):
    plt.subplot(4,5,1+i)
    sns.distplot(df[kolom[i]])
plt.show()

   labels_ = pd.read_csv("dengue_labels_train.csv")
   Df = pd.merge(df,labels_,on =
["city","year","weekofyear"])
   plt.figure(figsize=(20,10))
   sns.heatmap(df.corr(),annot=True)
   plt.show()

plt.figure(figsize=(15,8))

fig = sns.distplot(df["city"])

plt.show(fig)

plt.figure(figsize=(15,8))
fig = sns.distplot(df["station_precip_mm"])
plt.show(fig)
plt.figure(figsize=(15,8))
```

```
fig = sns.distplot(df["station_diur_temp_rng_c"])
plt.show(fig)
plt.figure(figsize=(15,8))
fig = sns.distplot(df["station_avg_temp_c"])
plt.show(fig)
plt.figure(figsize=(15,8))
fig = sns.distplot(df["reanalysis_min_air_temp_k"])
plt.show(fig)
plt.figure(figsize=(15,8))
fig = sns.distplot(df["year"])
plt.show(fig)
plt.figure(figsize=(15,8))
fig = sns.distplot(df["weekofyear"])
plt.show(fig)
```

```
filler ={
    "ndvi_ne" : df["ndvi_ne"].mean(),
    "ndvi_nw" : df["ndvi_nw"].mean(),
    "ndvi_se" : df["ndvi_se"].mean(),
    "ndvi_sw" : df["ndvi_sw"].mean(),
    "precipitation_amt_mm" :
df["precipitation_amt_mm"].median(),
    "reanalysis_air_temp_k" :
df["reanalysis_air_temp_k"].mean(),
    "reanalysis_avg_temp_k" :
df["reanalysis_avg_temp_k"].mean(),
    "reanalysis_dew_point_temp_k" :
df["reanalysis_dew_point_temp_k"].median(),
    "reanalysis_max_air_temp_k" :
df["reanalysis_max_air_temp_k"].median(),
    "reanalysis_min_air_temp_k" :
df["reanalysis_min_air_temp_k"].median(),
    "reanalysis_precip_amt_kg_per_m2" :
df["reanalysis_precip_amt_kg_per_m2"].median(),
```

```
    "reanalysis_relative_humidity_percent" :
df["reanalysis_relative_humidity_percent"].mean(),

    "reanalysis_sat_precip_amt_mm" :
df["reanalysis_sat_precip_amt_mm"].median(),

    "reanalysis_specific_humidity_g_per_kg" :
df["reanalysis_specific_humidity_g_per_kg"].median()
,

    "reanalysis_tdtr_k" :
df["reanalysis_tdtr_k"].median(),

    "station_avg_temp_c" :
df["station_avg_temp_c"].mean(),

    "station_diur_temp_rng_c" :
df["station_diur_temp_rng_c"].median(),

    "station_max_temp_c" :
df["station_max_temp_c"].mean(),

    "station_min_temp_c" :
df["station_min_temp_c"].mean(),

    "station_precip_mm"            :            df["
plt.figure(figsize=(15,15))

fig = df.boxplot()

fig.set_xticklabels(fig.get_xticklabels(),rotation=9
0)

plt.show()

station_precip_mm"].median()

}

df = df.fillna(filler)

df =
df[["city","station_precip_mm","station_diur_temp_rn
g_c","station_avg_temp_c","reanalysis_min_air_temp_k
","year","weekofyear","total_cases"]]


def recat_cat(x):

    if x == "sj":
```

```
        return 0
    elif x== "iq":
        return 1
kategorik = ["city"]
for i in kategorik:
    df[i] = df[i].apply(recat_cat)
df.head()
```

Machine Learning

```
from sklearn.model_selection import
train_test_split,GridSearchCV
from sklearn.preprocessing import
StandardScaler,MinMaxScaler,RobustScaler
from sklearn.metrics import
classification_report,confusion_matrix,accuracy_scor
e,f1_score,r2_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR


x = df.drop(columns="total_cases")
y = df["total_cases"]
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=.2,random_state=42)
x_train.shape,x_test.shape,y_train.shape,y_test.shap
e


RF = RandomForestRegressor()
RF.fit(x_train,y_train)
```

```
y_RF_train = RF.predict(x_train)

y_RF_test = RF.predict(x_test)

r2_RF_train = r2_score(y_train,y_RF_train)

r2_RF_test = r2_score(y_test,y_RF_test)

status_RF = []

print(f"Nilai r2 Score Random Forest : \n train :
{r2_RF_train} \n test : {r2_RF_test} ")

if (r2_RF_train-r2_RF_test) > 0.03 :

    print("Overfitting")

    status_RF.append("Overfitting")

elif (r2_RF_train-r2_RF_test) < -0.03 :

    print("UnderFitting")

    status_RF.append("Underfitting")

else :

    print("Just Right")

    status_RF.append("Just Right")


test_ =  pd.read_csv("dengue_features_test.csv")

test_ = test_.fillna(filler)

test_.isna().sum()

test_ =
test_[["city","station_precip_mm","station_diur_temp
_rng_c","station_avg_temp_c","reanalysis_min_air_tem
p_k","year","weekofyear"]]

for i in kategorik:

    test_[i] = test_[i].apply(recat_cat)

ytest_ = RF.predict(test_)


test_ = test_.drop(columns="week_start_date")
```

```
test_["total_cases"] = ytest_

RFModel =
test_[["city","year","weekofyear","total_cases"]]

RFModel


def recat_cat2(x):
    if x == 0:
        return "sj"
    elif x == 1:
        return "iq"
kategorik = ["city"]

for i in kategorik:
    RFModel[i] = RFModel[i].apply(recat_cat2)

RFModel.head()


RFModel["total_cases"] = [round(i) for i in
RFModel["total_cases"]]

RFModel.to_csv("RFModelFS.csv",index=False)
```

Lampiran 2 – Hasil Data Prediksi

	city	year	weekofyear	total_cases
1	sj	2008	18	6
2	sj	2008	19	8
3	sj	2008	20	4
4	sj	2008	21	6
5	sj	2008	22	9
6	sj	2008	23	13
7	sj	2008	24	14
8	sj	2008	25	12
9	sj	2008	26	18
10	sj	2008	27	26
11	sj	2008	28	37
12	sj	2008	29	34
13	sj	2008	30	38
14	sj	2008	31	47
15	sj	2008	32	58
16	sj	2008	33	61
17	sj	2008	34	25
18	sj	2008	35	57
19	sj	2008	36	56
20	sj	2008	37	55
21	sj	2008	38	60
22	sj	2008	39	30
23	sj	2008	40	23
24	sj	2008	41	56
25	sj	2008	42	36
26	sj	2008	43	31
27	sj	2008	44	33
28	sj	2008	45	30
29	sj	2008	46	37
30	sj	2008	47	33
31	sj	2008	48	28
32	sj	2008	49	32
33	sj	2008	50	23
34	sj	2008	51	27
35	sj	2008	52	23
36	sj	2009	1	7
37	sj	2009	2	24
38	sj	2009	3	20
39	sj	2009	4	19
40	sj	2009	5	17
41	sj	2009	6	16
42	sj	2009	7	17
43	sj	2009	8	16
44	sj	2009	9	13
45	sj	2009	10	9
46	sj	2009	11	10
47	sj	2009	12	9
48	sj	2009	13	8
49	sj	2009	14	9
50	sj	2009	15	7
51	sj	2009	16	6
52	sj	2009	17	5
53	sj	2009	18	7
54	sj	2009	19	7
55	sj	2009	20	7
56	sj	2009	21	6
57	sj	2009	22	13
58	sj	2009	23	10
59	sj	2009	24	16
60	sj	2009	25	22
61	sj	2009	26	17
62	sj	2009	27	14
63	sj	2009	28	36
64	sj	2009	29	35
65	sj	2009	30	39
66	sj	2009	31	54
67	sj	2009	32	62
68	sj	2009	33	56
69	sj	2009	34	107
70	sj	2009	35	24
71	sj	2009	36	74
72	sj	2009	37	25
73	sj	2009	38	30

74	sj	2009	39	64
75	sj	2009	40	74
76	sj	2009	41	75
77	sj	2009	42	76
78	sj	2009	43	69
79	sj	2009	44	35
80	sj	2009	45	27
81	sj	2009	46	30
82	sj	2009	47	27
83	sj	2009	48	31
84	sj	2009	49	29
85	sj	2009	50	27
86	sj	2009	51	25
87	sj	2009	52	38
88	sj	2010	53	7
89	sj	2010	1	16
90	sj	2010	2	13
91	sj	2010	3	23
92	sj	2010	4	18
93	sj	2010	5	14
94	sj	2010	6	12
95	sj	2010	7	12
96	sj	2010	8	14
97	sj	2010	9	13
98	sj	2010	10	13
99	sj	2010	11	10
100	sj	2010	12	9
101	sj	2010	13	14
102	sj	2010	14	8
103	sj	2010	15	6
104	sj	2010	16	7
105	sj	2010	17	16
106	sj	2010	18	16
107	sj	2010	19	15
108	sj	2010	20	21
109	sj	2010	21	20
110	sj	2010	22	19
111	sj	2010	23	29
112	sj	2010	24	30

113	sj	2010	25	22
114	sj	2010	26	33
115	sj	2010	27	36
116	sj	2010	28	33
117	sj	2010	29	74
118	sj	2010	30	26
119	sj	2010	31	45
120	sj	2010	32	62
121	sj	2010	33	63
122	sj	2010	34	23
123	sj	2010	35	110
124	sj	2010	36	65
125	sj	2010	37	79
126	sj	2010	38	76
127	sj	2010	39	68
128	sj	2010	40	72
129	sj	2010	41	104
130	sj	2010	42	29
131	sj	2010	43	25
132	sj	2010	44	27
133	sj	2010	45	34
134	sj	2010	46	25
135	sj	2010	47	29
136	sj	2010	48	30
137	sj	2010	49	28
138	sj	2010	50	20
139	sj	2010	51	18
140	sj	2011	52	28
141	sj	2011	1	18
142	sj	2011	2	13
143	sj	2011	3	19
144	sj	2011	4	14
145	sj	2011	5	14
146	sj	2011	6	14
147	sj	2011	7	17
148	sj	2011	8	18
149	sj	2011	9	13
150	sj	2011	10	9
151	sj	2011	11	9

152	sj	2011	12	9
153	sj	2011	13	9
154	sj	2011	14	8
155	sj	2011	15	6
156	sj	2011	16	8
157	sj	2011	17	7
158	sj	2011	18	7
159	sj	2011	19	8
160	sj	2011	20	7
161	sj	2011	21	7
162	sj	2011	22	20
163	sj	2011	23	16
164	sj	2011	24	17
165	sj	2011	25	44
166	sj	2011	26	32
167	sj	2011	27	19
168	sj	2011	28	34
169	sj	2011	29	88
170	sj	2011	30	42
171	sj	2011	31	74
172	sj	2011	32	28
173	sj	2011	33	66
174	sj	2011	34	28
175	sj	2011	35	91
176	sj	2011	36	65
177	sj	2011	37	69
178	sj	2011	38	24
179	sj	2011	39	78
180	sj	2011	40	78
181	sj	2011	41	78
182	sj	2011	42	67
183	sj	2011	43	28
184	sj	2011	44	55
185	sj	2011	45	25
186	sj	2011	46	27
187	sj	2011	47	29
188	sj	2011	48	31
189	sj	2011	49	31
190	sj	2011	50	29
191	sj	2011	51	20
192	sj	2012	52	20
193	sj	2012	1	19
194	sj	2012	2	20
195	sj	2012	3	20
196	sj	2012	4	19
197	sj	2012	5	18
198	sj	2012	6	16
199	sj	2012	7	13
200	sj	2012	8	16
201	sj	2012	9	13
202	sj	2012	10	10
203	sj	2012	11	10
204	sj	2012	12	8
205	sj	2012	13	9
206	sj	2012	14	8
207	sj	2012	15	7
208	sj	2012	16	6
209	sj	2012	17	6
210	sj	2012	18	9
211	sj	2012	19	7
212	sj	2012	20	13
213	sj	2012	21	9
214	sj	2012	22	13
215	sj	2012	23	17
216	sj	2012	24	16
217	sj	2012	25	32
218	sj	2012	26	34
219	sj	2012	27	31
220	sj	2012	28	36
221	sj	2012	29	36
222	sj	2012	30	39
223	sj	2012	31	43
224	sj	2012	32	60
225	sj	2012	33	74
226	sj	2012	34	67
227	sj	2012	35	28
228	sj	2012	36	66
229	sj	2012	37	64

230	sj	2012	38	66
231	sj	2012	39	76
232	sj	2012	40	78
233	sj	2012	41	72
234	sj	2012	42	76
235	sj	2012	43	74
236	sj	2012	44	77
237	sj	2012	45	77
238	sj	2012	46	37
239	sj	2012	47	69
240	sj	2012	48	30
241	sj	2012	49	29
242	sj	2012	50	32
243	sj	2012	51	21
244	sj	2013	1	16
245	sj	2013	2	18
246	sj	2013	3	18
247	sj	2013	4	18
248	sj	2013	5	20
249	sj	2013	6	18
250	sj	2013	7	15
251	sj	2013	8	12
252	sj	2013	9	12
253	sj	2013	10	8
254	sj	2013	11	10
255	sj	2013	12	8
256	sj	2013	13	7
257	sj	2013	14	8
258	sj	2013	15	11
259	sj	2013	16	9
260	sj	2013	17	5
261	iq	2010	26	4
262	iq	2010	27	2
263	iq	2010	28	3
264	iq	2010	29	3
265	iq	2010	30	4
266	iq	2010	31	3
267	iq	2010	32	4
268	iq	2010	33	4
269	iq	2010	34	4
270	iq	2010	35	5
271	iq	2010	36	5
272	iq	2010	37	5
273	iq	2010	38	5
274	iq	2010	39	7
275	iq	2010	40	5
276	iq	2010	41	7
277	iq	2010	42	7
278	iq	2010	43	5
279	iq	2010	44	6
280	iq	2010	45	10
281	iq	2010	46	11
282	iq	2010	47	9
283	iq	2010	48	8
284	iq	2010	49	10
285	iq	2010	50	14
286	iq	2010	51	9
287	iq	2011	52	11
288	iq	2011	1	15
289	iq	2011	2	13
290	iq	2011	3	14
291	iq	2011	4	15
292	iq	2011	5	13
293	iq	2011	6	15
294	iq	2011	7	15
295	iq	2011	8	18
296	iq	2011	9	13
297	iq	2011	10	9
298	iq	2011	11	11
299	iq	2011	12	10
300	iq	2011	13	8
301	iq	2011	14	7
302	iq	2011	15	4
303	iq	2011	16	5
304	iq	2011	17	5
305	iq	2011	18	6
306	iq	2011	19	4
307	iq	2011	20	4

308	iq	2011	21	3
309	iq	2011	22	3
310	iq	2011	23	3
311	iq	2011	24	2
312	iq	2011	25	3
313	iq	2011	26	1
314	iq	2011	27	1
315	iq	2011	28	3
316	iq	2011	29	3
317	iq	2011	30	3
318	iq	2011	31	4
319	iq	2011	32	4
320	iq	2011	33	5
321	iq	2011	34	4
322	iq	2011	35	4
323	iq	2011	36	6
324	iq	2011	37	6
325	iq	2011	38	10
326	iq	2011	39	10
327	iq	2011	40	9
328	iq	2011	41	10
329	iq	2011	42	5
330	iq	2011	43	15
331	iq	2011	44	12
332	iq	2011	45	16
333	iq	2011	46	8
334	iq	2011	47	11
335	iq	2011	48	6
336	iq	2011	49	15
337	iq	2011	50	18
338	iq	2011	51	17
339	iq	2012	52	8
340	iq	2012	1	13
341	iq	2012	2	16
342	iq	2012	3	13
343	iq	2012	4	10
344	iq	2012	5	13
345	iq	2012	6	13
346	iq	2012	7	14

347	iq	2012	8	15
348	iq	2012	9	12
349	iq	2012	10	8
350	iq	2012	11	9
351	iq	2012	12	10
352	iq	2012	13	7
353	iq	2012	14	8
354	iq	2012	15	6
355	iq	2012	16	5
356	iq	2012	17	5
357	iq	2012	18	5
358	iq	2012	19	4
359	iq	2012	20	4
360	iq	2012	21	4
361	iq	2012	22	2
362	iq	2012	23	3
363	iq	2012	24	3
364	iq	2012	25	3
365	iq	2012	26	3
366	iq	2012	27	3
367	iq	2012	28	3
368	iq	2012	29	4
369	iq	2012	30	3
370	iq	2012	31	5
371	iq	2012	32	3
372	iq	2012	33	4
373	iq	2012	34	4
374	iq	2012	35	5
375	iq	2012	36	4
376	iq	2012	37	5
377	iq	2012	38	7
378	iq	2012	39	6
379	iq	2012	40	9
380	iq	2012	41	5
381	iq	2012	42	8
382	iq	2012	43	15
383	iq	2012	44	8
384	iq	2012	45	8
385	iq	2012	46	16

386	iq	2012	47	10
387	iq	2012	48	7
388	iq	2012	49	5
389	iq	2012	50	13
390	iq	2012	51	12
391	iq	2013	1	17
392	iq	2013	2	13
393	iq	2013	3	13
394	iq	2013	4	14
395	iq	2013	5	25
396	iq	2013	6	15
397	iq	2013	7	15
398	iq	2013	8	15
399	iq	2013	9	17
400	iq	2013	10	10
401	iq	2013	11	11

402	iq	2013	12	8
403	iq	2013	13	9
404	iq	2013	14	9
405	iq	2013	15	5
406	iq	2013	16	6
407	iq	2013	17	4
408	iq	2013	18	5
409	iq	2013	19	4
410	iq	2013	20	5
411	iq	2013	21	3
412	iq	2013	22	3
413	iq	2013	23	3
414	iq	2013	24	3
415	iq	2013	25	3
416	iq	2013	26	3

Lampiran 3 - Nilai kesamaan penulisan dari Turnitin.