

**IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* UNTUK
MEMPREDIKSI HARGA SEWA AIRBNB MENGGUNAKAN METODE
RANDOM FOREST DAN PENERAPAN *WEB APPLICATION*
MENGGUNAKAN *FLASK*
(Studi Kasus: Airbnb Tokyo)**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Program Studi Statistika



**Maudi Mirqoatul Mafa'atih
16611098**

**JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2020**

**HALAMAN PERSETUJUAN PEMBIMBING
TUGAS AKHIR**

Judul : Implementasi *Artificial Intelligence* Untuk
Memprediksi Harga Sewa Airbnb
Menggunakan Metode *Random Forest* dan
Penerapan *Web Application* Menggunakan
Flask

Nama Mahasiswa : Maudi Mirqoatul Mafa'atih

Nomor Mahasiswa : 16611098

**TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK
DIUJIKAN**

Yogyakarta, 2 September 2020

Pembimbing

.....

(Arum Handini Primandari, S.Si., M.Si.)

HALAMAN PENGESAHAN

TUGAS AKHIR

IMPLEMENTASI ARTIFICIAL INTELLIGENCE UNTUK PREDIKSI HARGA SEWA AIRBNB MENGGUNAKAN METODE *RANDOM FOREST* DAN PENERAPAN *WEB APPLICATION* MENGGUNAKAN *FLASK*

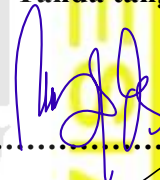
Nama Mahasiswa : Maudi Mirqoatul M
NIM : 16611098

TUGAS AKHIR INI TELAH DIUJIKAN
PADA TANGGAL 2 SEPTEMBER 2020

Nama Penguji

Tanda tangan

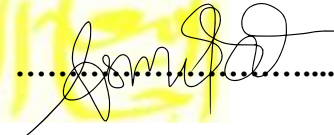
1. Mujiati Dwi Kartikasari, S.Si.,
M.Sc.



2. Tuti Purwaningsih, S.Stat., M.Si.



3. Arum Handini P., S.Pd.Si., M.Sc.



Mengetahui,
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



(Prof. Riyanto, S.Pd., M.Si., Ph.D.)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbil'aalamiin, puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada penulis sehingga penulis diberikan keimanan, kekuatan, kesehatan, kesabaran, kelancaran, serta keselamatan selama penyusunan tugas akhir ini hingga dapat terselesaikan. Shalawat serta salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad SAW beserta keluarga dan para pengikut- pengikutnya

Tugas akhir dengan judul “**Implementasi Artificial Intelligence Untuk Memprediksi Harga Sewa Airbnb Menggunakan Metode *Random Forest* dan Penerapan *Web Application* Menggunakan *Flask***” ini disusun sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang strata satu di Program Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia. Dengan segala kerendahan hati, penulis ucapkan terima kasih kepada pihak-pihak yang telah membantu hingga tugas akhir ini selesai, yaitu:

1. Prof. Riyanto, S.Pd., M.Si., Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Islam Indonesia.
2. Bapak Dr. Edy Widodo, S.Si., M.Si. selaku Ketua Jurusan Statistika beserta seluruh jajarannya.
3. Ibu Arum Handini Primandari, S.Si., M.Si. selaku dosen pembimbing yang sangat sabar dan berjasa membimbing dalam penyusunan tugas akhir.
4. Seluruh staff, pengajar Program Studi Statistika Universitas Islam Indonesia yang telah memberikan bekal ilmu dan bantuan kepada penulis.
5. Umi, abah, dan adik tercinta yang selalu mendoakan dan memberi dukungan kepada penulis baik moril maupun materiil.
6. Keluarga di Jogja arek kamar Hafsah lantai 3 yang selalu ada dan selalu menyemangati.
7. Teman-teman KKN unit 183, teman moga-jogja yang selalu menghibur dan memberi nasihat.

8. Sahabat seperjuangan Syinta, Rima, Gifa, Alfa, Lina, Farhan, dan Cindy yang selalu memberikan semangat, berbagi ilmu dan pengalaman.
9. Semua pihak yang turut membantu penulis dalam penyelesaian Tugas Akhir ini, penulis mengucapkan terima kasih semoga Allah SWT selalu memberikan rahmat dan anugerah-Nya kepada mereka semua tanpa henti. Aamiin ya robbal ,alamiin.

Semoga dukungan dan bantuan dari semua pihak senantiasa mendapat balasan dari Allah SWT. Penulis menyadari bahwa tugas akhir ini masih memiliki kekurangan di dalamnya. Hal tersebut karena keterbatasan ilmu dan pengetahuan yang dimiliki penulis semata. Penulis menerima kritik dan saran yang membangun demi perbaikan tugas akhir ini. Penulis berharap agar penelitian ini dapat bermanfaat dan memberikan khasanah pengetahuan bagi penulis, pembaca, maupun penelitian di masa depan

Wassalamualaikum Wr.Wb

Yogyakarta, 2 September 2020

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN PEMBIMBING	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR	ix
DAFTAR LAMPIRAN.....	x
PERNYATAAN.....	xi
ABSTRAK.....	xii
<i>ABSTRACT</i>	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB 2 TINJAUAN PUSTAKA.....	6
2.1 Kajian Pustaka.....	6
2.2 Kelebihan Penelitian.....	9
BAB 3 LANDASAN TEORI.....	10
3.1 Tokyo.....	10
3.2 Airbnb.....	11
3.3 CART (<i>Classification and Regression Tree</i>)	12
3.4 <i>Bootstrap Aggregating (Bagging)</i>	20
3.5 <i>Analisis Random Forest Regreesion</i>	20
3.5.1 <i>Pengertian Random Forest</i>	20
3.5.2 <i>Algoritma Random Forest</i>	21
3.6 <i>Variabel Importance</i>	22
3.7 <i>Akurasi Hasil Prediksi</i>	23

3.8	<i>Website Application (flask)</i>	24
3.9	<i>Label Encoder</i>	25
BAB 4 METODOLOGI PENELITIAN.....		27
4.1	Populasi dan sampel	27
4.2	Variabel Penelitian	27
4.3	Definisi Operasional Variabel	27
4.4	Metode Pengumpulan Data	34
4.5	Metode Penelitian.....	35
4.6	Perangkat Penelitian	35
4.7	Tahapan Penelitian	36
BAB 5 HASIL DAN PEMBAHASAN.....		38
5.1	<i>Preprocessing Data</i>	38
5.1.1	<i>Label Encoder</i>	38
5.1.2	Penentuan Data <i>Training</i> dan Data <i>Testing</i>	39
5.2	<i>Random Forest Regression</i>	40
5.2.1	Prediksi dan Validasi Model	40
5.2.2	Variabel <i>Importance</i>	43
5.3	Membuat <i>Web Application</i> Menggunakan Flask	44
5.3.1	<i>Export Model Random Forest</i>	44
5.3.2	<i>Import Package Flask</i>	45
5.3.3	Membuat HTML Tampilan Halaman <i>Website</i>	46
5.3.4	Menampilkan Halaman <i>Website</i>	47
5.3.5	Prediksi Harga Airbnb.....	48
BAB 6 PENUTUP		49
6.1.	Kesimpulan.....	49
6.2.	Saran.....	49
DAFTAR PUSTAKA		51
LAMPIRAN.....		54

DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya	8
Tabel 3.1 Contoh <i>Dataset</i> Pembentukan Pohon <i>Decision tree</i> Regresi	14
Tabel 3.2 Contoh Label <i>Encoder</i>	26
Tabel 4.1 Variabel Penelitian	27
Tabel 4.2 Definisi Operasional Variabel	27
Tabel 4.3 Nilai Variabel	28
Tabel 5.1 Konversi Variabel <i>Property type</i>	38
Tabel 5.2 Konversi Variabel <i>Room Type</i>	38
Tabel 5.3 Konversi Variabel <i>Neighbourhood</i>	39
Tabel 5.4 Konversi Variabel <i>Bed Type</i>	39
Tabel 5.5 Persentase Pembagian Data <i>Training</i> dan Data <i>Testing</i>	39
Tabel 5.6 Penentuan Jumlah Pohon	40
Tabel 5.7 Nilai Prediksi dan Nilai Aktual (Yen).....	42
Tabel 5.8 Hasil Akurasi	43



DAFTAR GAMBAR

Gambar 1.1 Jumlah Wisatawan pada 20 Kota Paling Banyak Dikunjungi Turis..	1
Gambar 3.1 Gambar Kota Tokyo	10
Gambar 3.2 Logo Airbnb	11
Gambar 3.3 Struktur Pohon Pada Cart	12
Gambar 4.1 Alur Penelitian	37
Gambar 5.1 Contoh <i>Output</i> Pohon <i>Random Forest Regression</i>	41
Gambar 5.2 Grafik Data Asli dengan Data Hasil Prediksi	41
Gambar 5.3 <i>Output</i> Variabel <i>Importance</i>	44
Gambar 5.4 <i>Export</i> Model <i>Random Forest</i>	45
Gambar 5.5 Direktori Folder Airbnb	47
Gambar 5.6 Menampilkan Alamat IP <i>Website</i>	47
Gambar 5.7 Tampilan <i>Website</i> Prediksi Harga Airbnb	47
Gambar 5.8 Input Nilai Variabel Prediktor	48
Gambar 5.9 <i>Output</i> Variabel Respon	48



DAFTAR LAMPIRAN

Lampiran 1. <i>Script</i> training dan ekspor model.....	54
Lampiran 2. <i>Script</i> App.py	56
Lampiran 3. <i>Script</i> halaman website	57
Lampiran 4. <i>Output</i> Pohon <i>Random Forest</i>	63



PERNYATAAN

Dengan ini saya menyatakan bahwa dalam tugas akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk tugas akhir. Tugas akhir ini diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 2 September 2020



Maudi Mirqoatul M

ABSTRAK

IMPLEMENTASI *ARTIFICIAL INTELIGENCE* UNTUK MEMPREDIKSI HARGA AIRBNB MENGGUNAKAN METODE *RAMDOM FOREST* DAN PENERAPAN *WEB APPLICATION* MENGGUNAKAN *FLASK*

(Studi Kasus: Airbnb Tokyo)

Maudi Mirqoatul Mafa'atih

Program Studi Statistika, Fakultas MIPA

Universitas Islam Indonesia

Airbnb merupakan salah satu *marketplace online* penyedia jasa penyewaan penginapan. Dalam perkembangannya, Airbnb sudah menjangkau hampir diseluruh dunia. Airbnb juga dapat digunakan para pemilik properti yang ingin menyewakan tempat tinggalnya kepada para wisatawan. Bagi pelanggan, memilih Airbnb dengan harga sewa yang ekonomis dan sesuai dengan fasilitas yang tersedia merupakan faktor yang penting. Sementara bagi pemilik properti, penentuan harga sewa yang kompetitif juga diperlukan untuk mendapatkan banyak pelanggan. Oleh karena itu diperlukan suatu model yang dapat digunakan untuk memprediksi harga sewa Airbnb dengan mempertimbangkan beberapa variabel yang diinputkan. Model prediksi tersebut dapat digunakan pelanggan maupun pemilik properti untuk membantu mengevaluasi harga yang ditawarkan dengan mengetahui informasi minimal tentang nilai properti yang tersedia. Dalam penelitian ini *Random Forest Regreesion* digunakan untuk melihat prediksi harga sewa Airbnb dengan variabel prediktor yang diinputkan yaitu Neighbourhood, Property Type, Room Type, Accommodates, Bathrooms, Bedrooms, Beds, Bed Type, Cleaning Fee, Guests Included, Extra People, Minimum Nights. *Random Forest Regression* merupakan salah satu metode *machine learning* yang dapat digunakan untuk menghasilkan model prediksi. Konsep perhitungan *Random Forest* yaitu dengan mengambil rata-rata dari sekian banyak hasil prediksi yang terbentuk sehingga diharapkan akan menghasilkan prediksi dengan tingkat akurasi yang tinggi. Hasil *testing* model *Random Forest Regreesion* mendapatkan nilai R^2 sebesar 0,682 atau 68,2% dengan nilai MSE 15389924.243 dan nilai MAPE sebesar 23.67% Kemudian model regresi yang sudah didapatkan akan dimasukkan kedalam *web application* menggunakan *library flask* pada python dan *deployment* menggunakan Heroku sehingga model dapat diakses secara luas.

Kata Kunci: Airbnb, *Random Forest Regreesion*, Flask

ABSTRACT

ARTIFICIAL INTELLIGENCE IMPLEMENTATION FOR AIRBNB RENT PREDICTION USING RAMDOM FOREST METHODS AND WEB APPLICATION USING FLASK

(Case Study : Airbnb Tokyo)

Maudi Mirqoatul Mafa'atiah

Department of Statistics, Faculty of Mathematics and Natural Science

Universitas Islam Indonesia

Airbnb is online marketplace for rental accommodation services which in development has reached almost all over the world. Airbnb can also be used by property owners who want to rent their residences to tourists. For customers, choosing Airbnb with economical rental price and in accordance with available facilities is an important factor. As for property owners, determining competitive rental rates is also needed to get many customers. So we need a model that can be used to predict Airbnb rental prices by considering several variables entered. The prediction model can be used by both the customer and the property owner to help evaluate the price offered by knowing the minimum information about the value of the available property. In this study Random Forest regression is used to see the prediction of Airbnb rental prices with predictor variables entered namely Neighborhood, Property Type, Room Type, Pool, Bathroom, Bedrooms, Beds, Bed Type, Cleaning Fee, Guests Included, Extra People, Minimum Nights. Random Forest regression is a machine learning method that can be used to produce predictive models. The concept of Random Forest calculation is to take the average of the many prediction results that are formed so that it is expected to produce predictions with a high degree of accuracy. The results of testing the Random Forest regression model get a R^2 value of 0.682 or 68.2%, MSE value 15389924.243 and MAPE value 23.67%. Then the regression model will be entered into a web application using the flask library in python and deployment using Heroku so the model can be accessed in general.

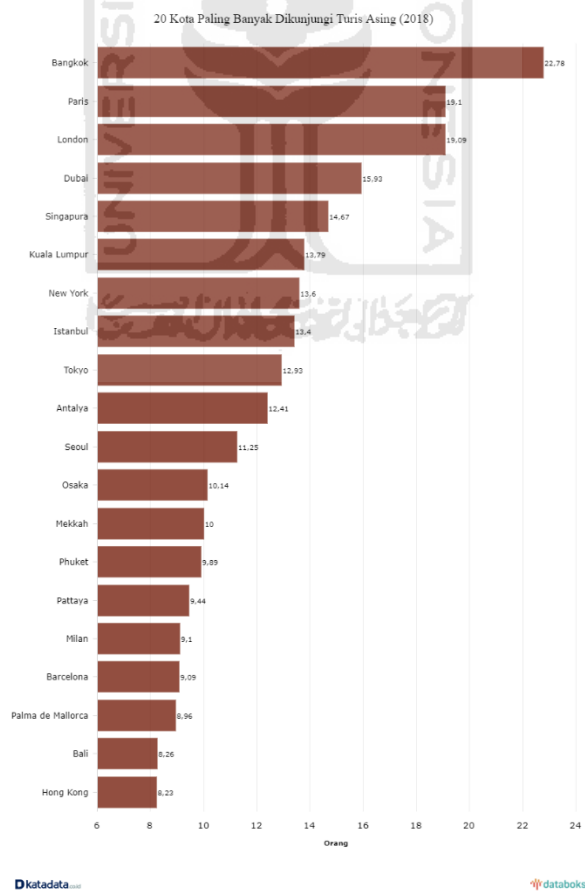
Keyword : *Airbnb, Random Forest Regression, Flask*

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Tokyo merupakan ibu kota Negara Jepang dengan luas wilayah metropolis sebesar 2.193.96 km². Menurut worldpopulationreview.com Tokyo merupakan salah satu kota dengan penduduk terpadat di dunia dengan jumlah penduduk lebih dari 37.39 juta jiwa (Worldpopulationreview.com, 2020). Selain padat karna banyaknya penduduk, Tokyo juga padat dengan banyaknya wisatawan yang berkunjung.

Situs online Databoks, Katadata Indonesia di tahun 2018 mengatakan bahwa Tokyo menjadi salah satu dari 20 kota yang banyak mendapat kunjungan turis asing dengan urutan ke-9 dan kunjungan sebanyak 12.93 juta wisatawan ke Tokyo pada tahun 2018.



Gambar 1.1 Jumlah Wisatawan pada 20 Kota Paling Banyak Dikunjungi Turis (Sumber : <https://databoks.katadata.co.id/>)

Tokyo menjadi kota metropolis yang paling menarik untuk dikunjungi di kawasan Asia. Tradisi dari zaman berabad lalu yang masih ada sampai sekarang dan berdampingan dengan elemen kebudayaan perkotaan dan teknologi yang terus berkembang sampai sekarang menjadi daya tarik tersendiri bagi wisatawan.

Dalam merencanakan liburan berkunjung ke Tokyo, salah satu aspek penting yang perlu diperhatikan adalah menentukan penginapan selama berada di Tokyo. Kawasan Tokyo yang luas dan beragam pilihan penginapan terkadang menjadi polemik tersendiri. Terlebih lagi *budget* yang harus disesuaikan mengingat Tokyo merupakan salah satu kota termahal didunia. Namun di zaman teknologi era digital seperti sekarang ini sudah banyak penyedia pelayanan akomodasi penginapan berbasis online salah satunya yaitu Airbnb. Airbnb bisa menjadi salah satu pilihan para wisatawan untuk dapat menemukan penginapan dengan harga yang relatif murah, dapat dipilih sesuai kebutuhan dan sudah banyak tersebar di seluruh penjuru dunia. Menurut website techcrunch.com pada 2014 Airbnb melaporkan lebih dari 800.000 akomodasi di seluruh dunia, dan telah melayani lebih dari 17 juta tamu di 600 kota dan 190 negara (Techcrunch.com, 2014) dan pada 2015 lebih dari 1,5 juta akomodasi terdaftar di 34.000 kota di seluruh dunia (Tang & Sangani, 2015).

Airbnb merupakan salah satu *marketplace* yang dapat digunakan untuk memesan penginapan secara online. Airbnb menyediakan berbagai macam pilihan penginapan dengan beragam harga yang dapat disesuaikan dengan kebutuhan. Selain itu, airbnb juga memfasilitasi para pemilik properti untuk dapat menyewakan tempat tinggalnya baik berupa rumah, kamar, apartemen maupun losmen. Seperti yang tercantum dalam website markey.id bahwa pada umumnya, pemilik properti yang disewakan tersebut memperoleh penghasilan tambahan melalui kerjasama dengan situs airbnb ini. Sementara bagi pengguna, layanan dari airbnb dapat dijadikan sebagai salah satu alternatif untuk memperoleh '*local experience*' dan akomodasi harga yang jauh lebih murah dibandingkan saat menginap di hotel (Markey.id, 2019).

Harga suatu penginapan yang ditawarkan pada situs airbnb sangat beragam. Keragaman tersebut berdasarkan pada antara lain jenis properti dan lokasi

penginapan tersebut. Penentuan harga merupakan salah satu faktor penentu yang dipertimbangkan wisatawan saat membandingkan tempat. Dalam penelitian Dewi Titiani Riski (2015), Kotler (2005:139) berpendapat bahwa kesalahan dalam menetapkan harga bukan hanya menjadi persoalan bagi produsen saja tetapi juga persoalan bagi konsumen, karena harga dapat mempengaruhi untuk membeli atau tidaknya konsumen terhadap produk tersebut (Riski, 2015). Oleh karena itu, sangatlah penting untuk menetapkan harga yang kompetitif di masing-masing wilayah. Penentuan harga sewa bagi pemilik properti membantu dalam penetapan harga internal karena akan menentukan jumlah pelanggan untuk tempat tersebut. Sedangkan bagi pelanggan atau wisatawan, penetapan harga membantu mengevaluasi harga yang ditawarkan pemilik properti dengan mengetahui informasi minimal tentang nilai properti yang tersedia.

Berdasarkan uraian sebelumnya, peneliti ingin melakukan prediksi harga sewa Airbnb untuk dapat membantu pemilik properti maupun pelanggan atau wisatawan. Penelitian tentang prediksi harga Airbnb sebelumnya sudah dilakukan oleh Emily Tang dan Kunal Sangani pada tahun 2015. Penelitian tersebut menggunakan dataset Airbnb San Francisco, dimana perusahaan Airbnb pertama kali dimulai dan menggunakan metode machine learning *SVM classifier*. Penelitian lain tentang prediksi harga Airbnb juga dilakukan oleh J. Li dan F. Biljecki pada tahun 2019. Dimana peneliti menggunakan dataset Airbnb Beijing dan menggunakan metode *Multiple Linear Regression*, *Gradient Boosting* dan *Random Forest Regressor*. Hasil penelitian menunjukkan model algoritma *Random Forest* menghasilkan nilai *R-Squared* yang paling tinggi yaitu 0.9178 dengan nilai RMSE sebesar 44295551.

Random Forest merupakan salah satu metode *machine learning* yang dapat digunakan untuk klasifikasi dan regresi. Metode ini merupakan sebuah *ensemble* (kumpulan) metode pembelajaran menggunakan pohon keputusan *base classifier* yang dibangun dan dikombinasikan (Primajaya & Sari, 2018). Pada metode *Random Forest* bukan hanya data sampel yang diambil secara acak untuk membentuk pohon klasifikasi, tetapi juga variabel independen diambil sebagian secara acak dan baru dipilih sebagai pemilah terbaik saat penentuan pemilah pohon,

sehingga diharapkan menghasilkan prediksi yang lebih akurat (Jatmiko, Padmadisastra, & Chadidjah, 2019).

Machine learning merupakan salah satu topik yang kian hari makin didengarkan dalam kalangan teknologi informasi. Sudah banyak perusahaan yang menggunakan *machine learning* untuk mendapatkan profit yang lebih baik. Dalam membangun sebuah model *machine learning* menggunakan scikit-learn, tensorflow, keras atau library python yang lain, seringkali para data *scientist* atau *machine learning engineer* lebih berfokus pada *Exploratory Data Analysis* (EDA) ataupun *feature engineering*, tetapi cenderung kurang memperhatikan tujuan utama dari membangun model tersebut, yaitu mengekstrak model atau membuatnya tersedia bagi sistem dan pengguna lain. Sehingga model *machine learning* tersebut dapat digunakan dan diterapkan dalam kehidupan sehari-hari.

Maka dari itu peneliti ingin mengangkat sebuah topik prediksi harga Airbnb menggunakan algoritma *machine learning Random Forest regressor*. Kemudian model regresi yang sudah didapatkan akan diterapkan kedalam sebuah *web application* sederhana menggunakan library python flask, sehingga akan mudah diakses baik untuk pemilik properti ataupun pelanggan guna memprediksi harga sewa Airbnb dengan memperhatikan beberapa variabel yang diinputkan.

1.2 Rumusan Masalah

Pada penelitian ini rumusan masalah yang ingin diangkat oleh peneliti adalah sebagai berikut:

1. Bagaimana memprediksi harga sewa Airbnb dengan metode *Random Forest regressor*?
2. Berapa tingkat akurasi hasil prediksi harga sewa Airbnb menggunakan metode *Random Forest regressor*?
3. Bagaimana membangun situs web untuk memprediksi harga sewa Airbnb dengan menggunakan metode *Random Forest regressor*?

1.3 Batasan Masalah

Permasalahan yang akan dibahas pada penelitian ini memiliki ruang lingkup yang luas, adapun batasan masalah sebagai berikut :

1. Penelitian ini menggunakan dataset Airbnb dari Tokyo City.

2. Algoritma yang digunakan dalam penelitian ini adalah *Random Forest regressor*.
3. Aplikasi yang digunakan adalah python dengan library sklearn dan flask.
4. Jumlah dataset yang digunakan berjumlah 8962.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian yang dilakukan adalah:

1. Memprediksi harga sewa Airbn dengan menggunakan metode *Random Forest regressor*.
2. Mengetahui tingkat akurasi yang didapatkan dari hasil prediksi harga sewa Airbnb menggunakan metode *Random Forest regressor*.
3. Membuat situs *web* untuk memprediksi harga sewa Airbnb dengan menggunakan metode *Random Forest regressor*.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian tugas akhir ini adalah :

1. Memberikan pengetahuan mengenai implementasi *Random Forest regressor*.
2. Memberikan pengetahuan terkait pengaplikasian metode *Random Forest regressor* kedalam suatu situs *web* dengan menggunakan library flask dan Heroku.
3. Membantu pemilik properti mempertimbangkan harga sewa yang akan ditawarkan kepada pelanggan.
4. Membantu pelanggan mengevaluasi harga sewa yang ditawarkan dengan mempertimbangkan beberapa fasilitas yang tersedia.
5. Memberi gambaran kepada pelanggan terkait faktor-faktor yang mempengaruhi harga sewa Airbnb.

BAB 2 TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian sebelumnya menjadi acuan untuk penelitian ini guna mengetahui hubungan dengan penelitian terdahulu dan menghindari unsur duplikasi dengan penelitian sebelumnya. Selain itu penelitian sebelumnya menunjukkan bahwa penelitian yang dilakukan mempunyai kebermanfaatan, sehingga dapat memberikan kontribusi pada perkembangan ilmu pengetahuan dan mampu memberikan solusi terhadap permasalahan aspek kehidupan dengan teknologi. Berikut ini adalah beberapa penelitian tentang implementasi *artificial intelligence* dengan metode *Random Forest* dan *flask*.

Penelitian dengan judul *Estimating Warehouse Rental Price Using Machine Learning Techniques* yang dilakukan oleh *Yixuan Ma* dan *Baoxiang Pan* pada tahun 2018. Penelitian ini bertujuan untuk mengestimasi harga rental yang berada di pasaran. Penelitian ini menggunakan empat metode *machine learning*, yaitu : *Linear Regression*, *Regression Tree*, *Random Forest Regreesion* dan *Gradient Boosting Regression Trees*. Data yang digunakan dalam penelitian diambil dan dikumpulkan dari situs web iklan baris di internet. Berdasarkan dataset yang digunakan, peneliti mengaplikasikan teknik *machine learning* yang digunakan untuk menghubungkan harga warehouse dengan beberapa fitur berkaitan yang relevan, seperti ukuran, lokasi, dan harga real estat disekitar. Hasil penelitian menunjukkan bahwa metode terbaik adalah *Random Forest* dengan koefisien korelasi 0,57 pada data *test*.

Penelitian mengenai prediksi harga Airbnb dengan judul *Airbnb Price Prediction Using Machine Learning and Sentiment Analysis* yang dilakukan oleh *Pouya Rezazadeh Kalehbasti* dan kawan-kawan pada tahun 2019. Penelitian ini menggunakan dataset Airbnb kota New York sebagai objeknya dan menggunakan tiga metode *machine learning*, yaitu *Support Vector Regression*, *K-means Clustering* dan *Neural Network* untuk membuat model prediksi. Hasil penelitian dengan jumlah data training sebanyak 39.980 dan data testing sebanyak 4.998 menunjukkan skor R^2 untuk model *Support Vector Regression* adalah 0.7768 (*train*

split) dan 0.6901 (*test split*), untuk model K-means adalah 0.6992 (*train split*) dan 0.6748 (*test split*) dan untuk model Neural Network adalah 0.7246 (*train split*) dan 0,6992 (*test split*).

Penelitian tentang prediksi harga Airbnb yang dilakukan oleh Yuanhang Luo, Xuanyu Zhou dan Yulian Zhou pada tahun 2019 menggunakan metode *Linear Regression*, *Random Forest*, *XGBoost* dan *Neural Network*. Pada penelitian ini data yang digunakan adalah Airbnb kota New York City, Paris dan Berlin. Hasil penelitian menunjukkan bahwa metode *Neural Network* menunjukkan nilai *R-Squared* yang lebih besar dibanding metode *machine learning* lain dengan skor *R-Squared* data *train* 0.769 dan *R-Squared* data *test* 0.741 untuk dataset Airbnb New York City. Skor *R-Squared* data *train* sebesar 0.762 dan *R-Squared* data *test* 0.716 untuk dataset Airbnb Paris. Peneliti juga membuat model prediksi dengan menggabungkan dataset Airbnb New York City dan Paris sehingga menghasilkan nilai *R-Squared* yang lebih besar yaitu 0.816 untuk data *train* dan 0.773 untuk data *test*.

Penelitian menggunakan dataset *Melbourne Airbnb* menggunakan metode *machine learning Linear Regression*, *Random Forest Regression*, *XGBoost Regression*, *Support Vector Regression* dan *Neural Network*. Tujuan penelitian untuk memprediksi harga *booking* dengan membandingkan beberapa metode *machine learning* yang digunakan. Peneliti menggunakan nilai *mean square error* (MSE) dan skor R^2 masing-masing algoritma untuk mendapatkan model yang optimal. Hasil penelitian menunjukkan nilai *R-Squared test* paling tinggi yaitu sebesar 0.6914 dengan menggunakan algoritma *Gradient Boosting*. Kemudian menggunakan algoritma *Random Forest* mendapatkan nilai *R-Squared test* sebesar 0.6539.

Penelitian implementasi analisis big data pada sewa Airbnb di Beijing yang dilakukan oleh J. Li dan F. Biljecki pada tahun 2019. Penelitian ini bertujuan membuat model untuk memprediksi harga Airbnb kemudian melakukan analisis sensitivitas untuk melihat perubahan harga sewa Airbnb seiring dengan perbedaan wilayahnya. Peneliti menggunakan tiga algoritma *machine learning* yaitu *Multiple Linear Regression*, *Gradient Boosting* dan *Random Forest* untuk membuat model

prediksi dengan mempertimbangkan nilai *R-Squared* dan RMSE yang dihasilkan untuk dipilih menjadi model terbaik dan dilanjutkan untuk analisis lebih lanjut. Hasil penelitian menunjukkan model algoritma *Random Forest* menghasilkan nilai *R-Squared* yang paling tinggi yaitu 0.9178 dengan nilai RMSE 44295551, di mana metode *Random Forest* kemudian dipilih sebagai model terbaik.

Tabel 2.1 Penelitian Sebelumnya

Peneliti	Judul Penelitian	Metode Penelitian	Hasil
Yixuan Ma dan Baoxiang Pan (2018)	<i>Estimating Warehouse Rental Price Using Machine Learning Techniques</i>	Linear Regression, Regression Tree, <i>Random Forest</i> Regression dan Gradient Boosting Regression Trees	Hasil penelitian menunjukkan metode terbaik adalah <i>Random Forest</i> dengan koefisien korelasi 0,57 pada data test.
Pouya Rezazadeh Kalebasti dkk (2019)	<i>Airbnb Price Prediction Using Machine Learning and Sentiment Analysis</i>	Support Vector Regression, K-means Clustering dan Neural Network	Hasil penelitian dengan jumlah data training sebanyak 39.980 dan data testing sebanyak 4.998 menunjukkan skor R^2 untuk model <i>Support Vector Regression</i> adalah 0.7768 (<i>train split</i>) dan 0.6901 (<i>test split</i>) sehingga dipilih sebagai model terbaik untuk memprediksi harga Airbnb New York City.
Yuanhang Luo, Xuanyu Zhou dan Yulian Zhou (2019)	<i>Predicting Airbnb Listing Price Across Different Cities</i>	Linear Regression, <i>Random Forest</i> , XGBoost dan Neural Network	Hasil penelitian menunjukkan metode <i>Neural Network</i> mendapatkan nilai <i>R-Squared</i> yang lebih besar dibanding metode <i>machine learning</i> lain dengan skor <i>R-Squared</i> data <i>train</i> 0.769 dan <i>R-Squared</i> data <i>test</i> 0.741 untuk dataset Airbnb New York City. Skor <i>R-Squared</i> data <i>train</i> sebesar 0.762 dan <i>R-Squared</i> data <i>test</i> 0.716

Peneliti	Judul Penelitian	Metode Penelitian	Hasil
			untuk dataset Airbnb Paris. Peneliti juga membuat model prediksi dengan menggabungkan dataset Airbnb New York City dan Paris sehingga menghasilkan nilai <i>R-Squared</i> yang lebih besar yaitu 0.816 untuk data <i>train</i> dan 0.773 untuk data <i>test</i> .
Tiancheng Cai, Kevin Han dan Han Wu (2018)	<i>Melbourne Airbnb Price Prediction</i>	Linear Regression, <i>Random Forest</i> Regression, XGBoost Regression, Support Vector Regression dan Neural Network	Hasil penelitian menunjukkan nilai <i>R-Squared test</i> paling tinggi yaitu sebesar 0.6914 dengan menggunakan algoritma <i>Gradient Boosting</i> .
J. Li, F. Biljecki (2019)	<i>The Implementation Of Big Data Analysis In Regulating Online Short-Term Rental Business: A Case Of Airbnb In Beijing</i>	Multiple Linear Regression, Gradient Boosting dan <i>Random Forest</i>	Hasil penelitian menunjukkan model algoritma <i>Random Forest</i> menghasilkan nilai <i>R-Squared</i> yang paling tinggi yaitu 0.9178 dengan nilai RMSE sebesar 44295551.

2.2 Kelebihan Penelitian

Kelebihan penelitian ini dari penelitian sebelumnya adalah objek prediksi yaitu Kota Tokyo dan pengaplikasian model prediksi kedalam flask atau *website application* sehingga mempermudah siapapun untuk dapat mengaksesnya.

BAB 3 LANDASAN TEORI

3.1 Tokyo

Tokyo merupakan ibu kota Jepang dengan luas wilayah metropolis sebesar 2.193.96 km². Tokyo merupakan salah satu kota dengan penduduk terpadat di dunia dengan jumlah penduduk lebih dari 37.39 juta jiwa (Worldpopulationreview.com, 2020). Kawasan Tokyo yang lebih besar terdiri dari Tokyo dan tiga wilayah tetangganya yaitu Saitama, Kanagawa dan Chiba. Tokyo terdiri dari kawasan administratif yang lebih kecil, yaitu daerah khusus dan kota madya. Wilayah pusatnya terbagi menjadi 23 daerah khusus serta daerah Tama sebelah barat yang mencakup 26 kota besar, 3 kota kecil dan 1 desa. Kedua puluh tiga daerah tersebut membentuk wilayah yang memanjang sekitar 90 km dari timur ke barat dan 25 km dari utara ke selatan. Kemudian Kepulauan Izu dan Kepulauan Ogasawara juga secara administratif menjadi bagian dari kota Tokyo



(sumber : <https://id.wikipedia.org/wiki/Tokyo>)

Gambar 3.1 Gambar Kota Tokyo

Tokyo menjadi kota metropolis yang paling menarik untuk dikunjungi di kawasan Asia. Tradisi dari zaman berabad lalu yang masih ada sampai sekarang dan berdampingan dengan elemen kebudayaan perkotaan yang terus berkembang menjadi daya tarik tersendiri untuk wisatawan.

3.2 Airbnb

Airbnb atau dikenal dari singkatan *air bed and breakfast* merupakan salah satu *marketplace* untuk dapat memungkinkan penggunanya menyewa kamar, rumah, apartemen atau hunian lainnya. Airbnb juga dapat digunakan bagi pemilik properti yang ingin menyewakan tempat tinggalnya baik berupa rumah, apartemen, maupun kamar pribadi dalam jangka waktu tertentu. Singkatnya, airbnb menjadi penghubung pihak yang akan menyewa dengan pihak yang menyewakan.

Airbnb berdiri mulai tahun 2008 yang dipelopori oleh Brian Chesky dan Joe Gebbia. Dalam 6 tahun, Airbnb telah membantu lebih dari 15 juta tamu melalui lebih dari 800.000 listing di 34.000 kota dan 190 negara dan terus berkembang sampai sekarang.



Gambar 3.2 Logo Airbnb

Logo Airbnb merupakan abstraksi dari empat prinsip yaitu *People*, *Places*, *Love*, dan *Airbnb* yang digabung kemudian menjadi satu membentuk huruf "A". Dalam situs online lifestyle.okezone.com dijelaskan bahwa makna masing-masing empat prinsip tersebut adalah (Lifestyle.okezone.com, 2018):

- *People* atau membentuk kepala orang, menyiratkan pengguna situs.
- *Places* atau tanda untuk lokasi di peta menyiratkan tempat di mana rumah atau apartemen berada.
- *Love* merupakan tanda cinta.
- Kemudian ketiga elemen tersebut digabungkan untuk membentuk elemen lain yang membentuk huruf A yaitu Airbnb.

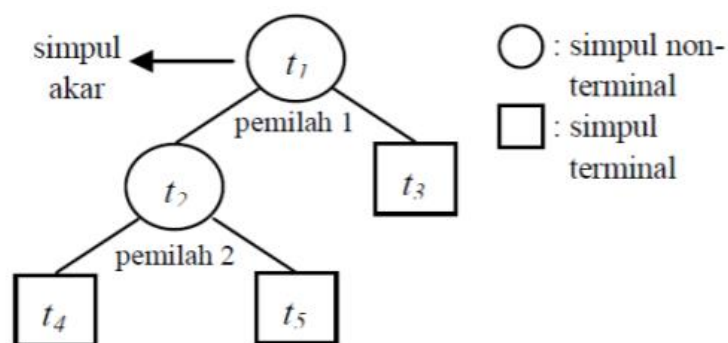
Visi dan misi Airbnb seperti yang tertera dalam logo adalah “*belong anywhere*”, yang artinya Airbnb ingin menghubungkan orang-orang yang membutuhkan akomodasi dengan mereka yang menyediakan akomodasi dan juga menciptakan dunia dimana bisa ikut merasakan menjadi bagian dari masyarakat, tempat dan cinta dimanapun mereka berada.

3.3 CART (Classification and Regression Tree)

Classification and Regression Trees (CART) merupakan salah satu metode atau algoritma dari teknik pohon keputusan (*decision tree*). Metode ini dikembangkan oleh Leo Breiman, Jerome H. Friedman, Richard A. Olshen, dan Charles J. Stone (Sumartinidan & Purnami, 2015). Metode CART dapat digunakan untuk menggambarkan hubungan antara variabel respon (dependen) dengan variabel predictor (variabel independen). CART akan menghasilkan pohon klasifikasi jika variabel respon mempunyai skala kategorik dan akan menghasilkan pohon regresi jika variabel respon berupa data kontinu.

Konsep pembentukan pohon keputusan dalam CART yaitu data *learning* akan membentuk suatu ruang yang disebut simpul (*node*) menjadi dua simpul anak dan setiap simpul anak dapat dipilah lagi menjadi dua simpul anak berikutnya, begitu seterusnya dan berhenti jika telah mendapatkan sekelompok observasi yang relatif homogen (Aryo, Padmadisastra, & Chadidjah, 2018).

Struktur pembentukan pohon dalam CART dapat digambarkan secara sederhana seperti pada **gambar 3.3** berikut (Budi A, 2015) :



Gambar 3.3 Struktur Pohon Pada Cart

Pada **gambar 3.3**, Pohon yang dibangun dibagi secara rekursif dari proses *pohon* pemilihan pada suatu gugus data kelas yang sama. Setiap pemilah (*split*)

memilah simpul non terminal menjadi dua simpul yang saling lepas. Kemudian hasil prediksi respons pada amatan akan terdapat pada simpul terminal (Budi A, 2015).

Proses pembentukan pohon dalam CART secara garis besar terbagi menjadi 4 bagian, yaitu (J. Lewis, 2000):

1. Langkah pertama terdiri dari pembentukan pohon. Pohon dibangun dengan pemisahan simpul secara rekursif. Setiap simpul yang dihasilkan akan menjadi *predictor* untuk simpul anak selanjutnya dan terus berulang sampai mustahil untuk dilanjutkan.
2. Langkah kedua yaitu menghentikan proses pembangunan pohon. Pada titik ini pohon dengan maksimal jumlah level yang sudah ditentukan sebelumnya telah diproduksi, dimana pembentukan simpul terminal ini mungkin sudah cocok dengan informasi yang terkandung di dalam *learning* dataset.
3. Langkah ketiga terdiri dari pemangkasan pohon yang menghasilkan simpul urutan pohon yang lebih sederhana.
4. Langkah keempat adalah penentuan pohon optimal, dimana pohon yang terbentuk sesuai dengan informasi dalam *learning* dataset.

Algoritma inti yang digunakan untuk membangun sebuah pohon keputusan dapat dihitung dengan berbagai cara, diantaranya menggunakan algoritma pengurangan standar deviasi atau *Standard Deviation Reduction* (SDR). Algoritma ini menggunakan rumus standar varians untuk memilih pemisah terbaik. Pemisahan dengan varian yang lebih rendah dipilih sebagai kriteria untuk *split* dataset. Algoritma SDR ini bisa digunakan untuk membentuk pohon keputusan dengan variabel target berupa data kontinu atau masalah regresi.

Berikut merupakan contoh pembentukan pohon keputusan bentuk regresi dengan menggunakan algoritma *Standard Deviation Reduction* (SDR). Misalnya, dari dataset berikut akan mempertimbangkan berapa jam seseorang dapat bermain suatu permainan di lapangan berdasarkan parameter kondisi cuaca. Dari parameter yang diberikan, akan dibentuk sebuah pohon keputusan untuk melihat berapakah lamanya pemain dapat bermain dengan melihat empat parameter terkait tersebut (Sayad, 2010).

Tabel 3.1 Contoh Dataset Pembentukan Pohon *Decision tree* Regresi

Outlook	Temp	Humidity	Windy	Hours played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	46
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30

Algoritma ID3 dapat digunakan untuk membangun pohon keputusan regresi dengan mengganti *Information Gain* dengan *Standard Deviation Reduction* (SDR). Berikut adalah langkah langkahnya:

a. Standar deviasi untuk 1 variabel

Hours played
26
30
48
46
62
23
43
36
38
46
48
52
44
30

$$\text{Count} = n = 14$$

$$\text{Average} = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$\rightarrow \text{Standard Deviation} = S = \sqrt{\frac{\sum (x - \bar{x})^2}{n}} = 9.32$$

$$\text{Coefficient of Variation} = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

- Standar deviasi (S) untuk membangun pohon percabangan.
- *Coefficient of Deviation* (CV) digunakan untuk menjadi acuan kapan untuk menghentikan percabangan.

- *Average (Avg)* digunakan untuk nilai yang berada di simpul terminal atau simpul daun.

b. Standar deviasi untuk 2 variabel (variabel target dan prediktor)

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

$$S(\text{Hours, Outlook}) = P(\text{Sunny}) * S(\text{Sunny}) + P(\text{Overcast}) * S(\text{Overcast}) + P(\text{Rainy}) * S(\text{Rainy})$$

$$= \left(\frac{4}{14}\right) * 3.49 + \left(\frac{5}{14}\right) * 7.78 + \left(\frac{5}{14}\right) * 10.87$$

$$= 7.66$$

Pengurangan Standar Deviasi

Langkah 1:

Menghitung standar deviasi dari variabel target.

$$\text{Standar deviasi (Hours played)} = 9.32$$

Langkah 2:

Menghitung standar deviasi dari masing-masing variabel prediktor. Nilai standar deviasi yang dihasilkan dikurangkan dengan nilai standar deviasi sebelum pemisahan. Hasilnya adalah pengurangan standar deviasi.

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
		SDR=1.66

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
		SDR=0.17

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
		SDR=0.28

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
		SDR=0.29

$$SDR(T, X) = S(T) - S(T, X)$$

$$\begin{aligned}
 SDR(Hours, Outlook) &= S(Hours) - S(Hours, Outlook) \\
 &= 9.32 - 7.66 \\
 &= 1.66
 \end{aligned}$$

Langkah 3:

Variabel dengan nilai pengurangan standar deviasi terbesar dipilih untuk menjadi simpul keputusan.

		<i>Hours Played (StDev)</i>
<i>Outlook</i>	<i>Overcast</i>	3.49
	<i>Rainy</i>	7.78
	<i>Sunny</i>	10.87
		→ SDR=1.66

Langkah 4:

- a. Dataset dibagi berdasarkan variabel yang terpilih untuk menjadi simpul keputusan. Proses ini berjalan secara rekursif hingga semua data terproses.

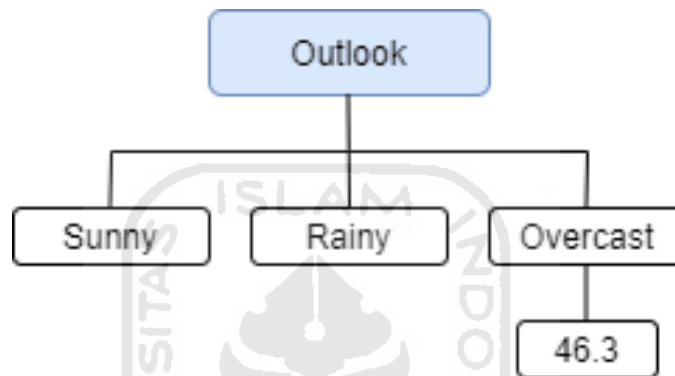
Outlook		Outlook	Temp	Humidity	Windy	Hours played
		Overcast	Overcast	Mild	High	True
		Overcast	Hot	Normal	False	44
		Overcast	Hot	High	False	46
		Overcast	Cool	Normal	True	43
		Rainy	Hot	High	False	25
		Rainy	Hot	High	True	30
		Rainy	Mild	High	False	35
		Rainy	Cool	Normal	False	38
		Rainy	Mild	Normal	True	48
		Sunny	Mild	Normal	False	46
		Sunny	Mild	High	False	45
		Sunny	Cool	Normal	False	62
		Sunny	Cool	Normal	True	23
		Sunny	Mild	High	True	30

Dalam pembentukan pohon, dibutuhkan kriteria untuk pemberhentian penumbuhan pohon. Misalnya, pohon akan berhenti ketika nilai CV untuk cabang menjadi lebih kecil dari 10% atau ketika terlalu sedikit contoh (n) tetap berada di cabang (misal 3).

- b. Subset *Overcast* tidak memerlukan pemisahan lebih lanjut karena nilai *CV* nya kurang dari batas minimal pemutusan simpul.

Outlook - Overcast

		<i>Hours Played (StDev)</i>	<i>Hours Played (AVG)</i>	<i>Hours Played (CV)</i>	<i>Count</i>
<i>Outlook</i>	<i>Overcast</i>	3.49	46.3	8%	4
	<i>Rainy</i>	7.78	35.2	22%	5
	<i>Sunny</i>	10.87	39.2	28%	5



- c. Cabang *Sunny* memiliki *CV* 28% lebih besar dari ambang batas yaitu 10% sehingga membutuhkan pemisahan lebih lanjut. Variabel *Windy* dipilih sebagai simpul terbaik setelah *Outlook* karena memiliki nilai **SDR** terbesar.

Outlook - Sunny

Temp	Humidity	Windy	Hours played
Mild	Normal	False	46
Mild	High	False	45
Cool	Normal	False	62
Cool	Normal	True	23
Mild	High	True	30
			S = 10.87
			AVG = 39.2
			CV = 28%

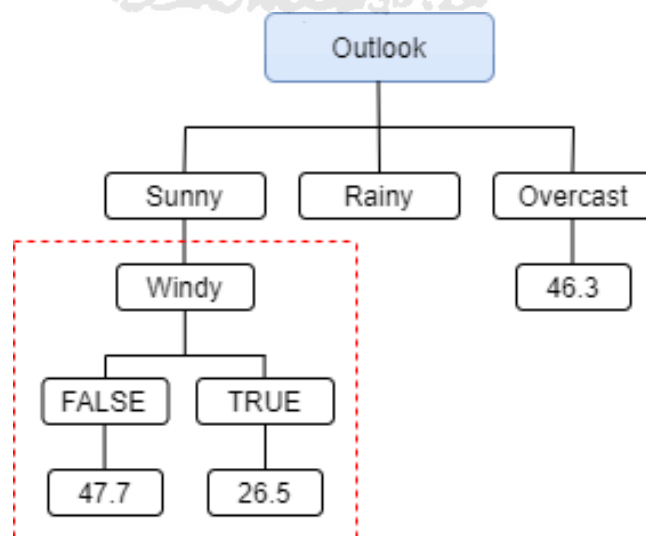
		<i>Hours Played (StDev)</i>	<i>Count</i>
<i>Temp.</i>	<i>Cool</i>	14.50	2
	<i>Mild</i>	7.32	3
$\mathbf{SDR} = 10.87 - \left(\left(\frac{2}{5} \right) * 14.5 + \left(\frac{3}{5} \right) * 7.32 \right) = 0.678$			

		<i>Hours Played</i> (<i>StDev</i>)	<i>Count</i>
<i>Windy</i>	<i>False</i>	3.09	3
	<i>True</i>	3.50	2
$\text{SDR} = 10.87 - \left(\left(\frac{3}{5} \right) * 3.09 + \left(\frac{3}{5} \right) * 3.5 \right) = 7.62$			

		<i>Hours Played</i> (<i>StDev</i>)	<i>Count</i>
<i>Humidity</i>	<i>High</i>	7.50	2
	<i>Normal</i>	12.50	3
$\text{SDR} = 10.87 - \left(\left(\frac{2}{5} \right) * 7.5 + \left(\frac{3}{5} \right) * 12.5 \right) = 0.370$			

Jumlah titik data untuk kedua cabang (*false* dan *true*) sama atau kurang dari 3, maka percabangan dihentikan dan menetapkan rata-rata untuk setiap cabang simpul daun terkait.

Temp	Humidity	Windy	Hours played
Mild	Normal	False	46
Mild	High	False	45
Cool	Normal	False	62
Cool	Normal	True	23
Mild	High	True	30



- d. Selanjutnya cabang *Rainy* memiliki CV 22% yang melebihi ambang batas yaitu 10%. Cabang ini membutuhkan percabangan lebih lanjut.

Outlook – Rainy

Temp	Humidity	Windy	Hours played
Hot	High	False	25
Hot	High	True	30
Mild	High	False	35
Cool	Normal	False	38
Mild	Normal	True	48
			S = 7.78
			AVG = 35.2
			CV = 22%

		<i>Hours Played (StDev)</i>	<i>Count</i>
<i>Temp.</i>	<i>Cool</i>	0	1
	<i>Hot</i>	2.5	2
	<i>Mild</i>	6.5	2
$\mathbf{SDR} = 7.78 - \left(\left(\frac{1}{5} \right) * 0 + \left(\frac{2}{5} \right) * 2.5 + \left(\frac{2}{5} \right) * 6.5 \right) = \mathbf{4.18}$			

		<i>Hours Played (StDev)</i>	<i>Count</i>
<i>Windy</i>	<i>False</i>	5.6	3
	<i>True</i>	9.0	2
$\mathbf{SDR} = 7.78 - \left(\left(\frac{3}{5} \right) * 5.6 + \left(\frac{2}{5} \right) * 9.0 \right) = \mathbf{0.82}$			

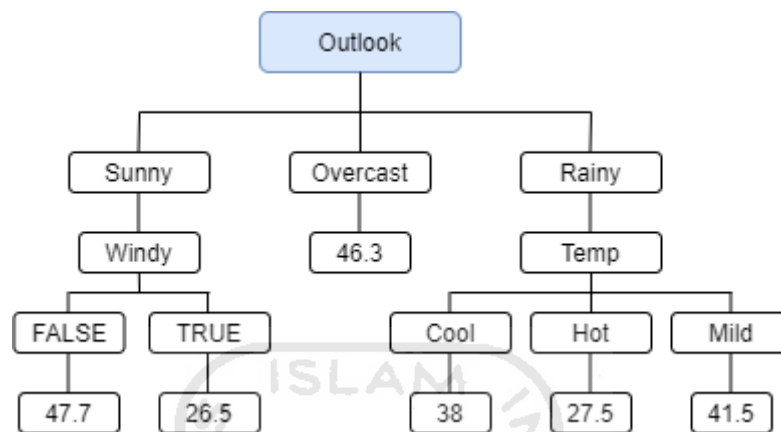
		<i>Hours Played (StDev)</i>	<i>Count</i>
<i>Humidity</i>	<i>High</i>	4.1	3
	<i>Normal</i>	5.0	2
$\mathbf{SDR} = 7.78 - \left(\left(\frac{3}{5} \right) * 4.1 + \left(\frac{2}{5} \right) * 5.0 \right) = \mathbf{3.32}$			

Jumlah titik data untuk ketiga cabang (*cool*, *hot* dan *mild*) sama atau kurang dari 3, maka percabangan dihentikan dan menetapkan rata-rata untuk setiap cabang simpul daun terkait.

Temp	Hours played
Cool	38
Hot	25
Hot	30
Mild	35

Temp	Hours played
Mild	48

Sehingga dari hasil perhitungan pengurangan standar deviasi yang telah dilakukan akan menghasilkan pohon keputusan seperti berikut:



3.4 Bootstrap Aggregating (Bagging)

Bagging merupakan singkatan dari *Bootstrap Aggregating*, yaitu metode prediktor yang menghasilkan beberapa versi prediktor kemudian menggunakannya untuk mendapatkan prediktor gabungan sehingga dapat memperbaiki kualitas prediksi yang dihasilkan.

Metode *Bagging* merupakan penyempurnaan dari metode CART yaitu dengan menggabungkan banyak nilai dugaan menjadi satu nilai dugaan sehingga sangat membantu terutama mengatasi sifat ketidakstabilan *tree* klasifikasi dan regresi tunggal (Wibowo, 2016).

Konsep dasar dalam metode *Bagging* adalah dengan menggunakan *resampling* acak dan pengembalian pada dataset awal sehingga diperoleh suatu dataset baru. Dataset baru tersebut kemudian digunakan untuk membangkitkan pohon klasifikasi dengan banyak versi. Pohon klasifikasi dari setiap versi kemudian digabungkan untuk memperoleh prediksi akhir (Aryo, Padmadisastra, & Chadidjah, 2018).

3.5 Analisis Random Forest Regression

3.5.1 Pengertian Random Forest

Menurut Vrushali Y Kulkarni dalam penelitian Aji Primajaya disebutkan bahwa *Random Forest* merupakan salah satu metode yang dapat digunakan untuk

klasifikasi dan regresi. Metode ini merupakan sebuah *ensemble* (kumpulan) metode pembelajaran menggunakan pohon keputusan *base classifier* yang dibangun dan dikombinasikan (Primajaya & Sari, 2018).

Metode *Random Forest* merupakan bagian dari metode *decision tree* dan bagian dari pengembangan metode CART (*Classification and Regression Tree*). Menurut salah satu pengusul metode *Random Forest* yaitu Breiman, dalam *Random Forest bagging* diperluas dan dikombinasikan dengan pengacakan variabel input yang digunakan ketika mempertimbangkan variabel kandidat untuk membagi node internal. Pada setiap node algoritma *Random Forest* memilih bagian acak diantara semua variabel, kemudian menentukan pemisahan terbaik atas variabel-variabel tersebut (Gilles Louppe, 2013).

Random Forest dikembangkan dengan gagasan perlu adanya penambahan layer pada proses *resampling* acak pada *bagging*. Oleh karena itu, bukan hanya data sampel yang diambil secara acak untuk membentuk pohon klasifikasi, tetapi juga variabel independen diambil sebagian secara acak dan baru dipilih sebagai pemilah terbaik saat penentuan pemilah pohon, sehingga diharapkan menghasilkan prediksi yang lebih akurat (Aryo, Padmadisastra, & Chadidjah, 2018).

3.5.2 Algoritma *Random Forest*

Dalam algoritma *Random Forest* membutuhkan dua parameter yaitu jumlah *tree* dan jumlah atribut yang digunakan untuk memperbanyak *tree* (Annisa, 2019). Penggunaan algoritma dalam *Random Forest* dilakukan dalam tahap sebagai berikut (Liaw & Wiener, 2001) :

1. Buat n_{tree} sampel *bootstrap* dari data *training*.
2. Untuk setiap hasil sampel *bootstrap*, susun pohon klasifikasi dengan pemilah terbaik melalui proses pemangkasan dengan mengulangi setiap tahap pada terminal simpul.
3. Ulangi langkah 1 dan 2 sebanyak k kali sehingga mendapatkan k buah pohon acak.
4. Memprediksi data baru dengan menggabungkan hasil prediksi pohon n_{tree} .

Dalam *Random Forest*, setiap kandidat peubah penjelas dalam pembentukan pohon merupakan hasil pemilihan secara acak. Sehingga

menghasilkan kumpulan pohon yang berbeda bentuk dan ukuran. Hasil yang diharapkan dalam setiap pembentukan pohon adalah setiap kumpulan pohon tersebut memiliki nilai korelasi yang kecil antar pohonnya. Sehingga akan mengakibatkan nilai ragam dugaan menjadi kecil atau lebih kecil dibandingkan hasil ragam dugaan algoritma *bagging* (Wibowo, 2016).

3.6 Variabel *Importance*

Variabel *importance* adalah ukuran kepentingan setiap variabel respon atau variabel prediktor dengan variabel lainnya. Dalam *Random Forest* ukuran kepentingan ini diketahui dengan menghitung berapa kali setiap variabel dipilih oleh semua pohon individu dalam suatu *ensemble* (Strobl, Boulesteix, Zeileis, & Hothorn, 2007). Untuk setiap pohon keputusan, variabel *importance* dihitung menggunakan *Gini Importance* dengan asumsi hanya dua simpul penurunan:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \quad (1)$$

dengan:

- ni_j : pentingnya simpul j
- w_j : jumlah sampel yang mencapai simpul j
- C_j : nilai *impurity* dari simpul j
- $left_j$: anak simpul dari *split* kiri pada simpul j
- $right_j$: anak simpul dari *split* kanan pada simpul j

Kemudian ukuran kepentingan untuk setiap *feature* pada pohon keputusan dapat dihitung sebagai berikut:

$$fi_i = \frac{\sum_{j: \text{simpul } j \text{ splits on feature } i} ni_j}{\sum_{\text{kesemua simpul}} ni_k} \quad (2)$$

dengan:

- fi_i : pentingnya *feature* i
- ni_j : pentingnya simpul j

Normalisasi nilai menjadi antara 0 dan 1 dengan membagi jumlah semua nilai ukuran kepentingan menggunakan rumus:

$$normfi_i = \frac{fi_i}{\sum_{j \in \text{all features}} fi_j} \quad (3)$$

Sementara pada *Random Forest*, ukuran kepentingan dihitung dengan membagi jumlah nilai kepentingan pada setiap pohon dengan jumlah total pohon:

$$RFfi_i = \frac{\sum_{j \in \text{alltrees}} \text{norm}f_{ij}}{T} \quad (4)$$

dengan:

- $RFfi_i$: nilai ukuran kepentingan i dari semua pohon dalam model *Random Forest*
- $\text{norm}f_{ij}$: normalisasi nilai ukuran kepentingan untuk i pada pohon j
- T : jumlah pohon

Secara umum tujuan variabel *importance* ini adalah untuk mengetahui dampak dari masing-masing variabel prediktor secara individual maupun dalam interaksi multivariat dengan variabel prediktor lainnya. Dalam *variabel importance* juga dapat digunakan sebagai kriteria pemilihan variabel dengan mengidentifikasi prediktor yang relevan dari sejumlah variabel kandidat.

3.7 Akurasi Hasil Prediksi

Dalam membangun sebuah *regressor* penting untuk mengetahui kualitas *regressor* tersebut. Hasil suatu *regressor* dapat dilihat performanya dengan memperhatikan beberapa parameter pengukuran, seperti dengan melihat nilai *Error* yang dihasilkan oleh jaringan. Nilai error disini didefinisikan sebagai perbedaan nilai aktual dengan nilai prediksi yang dihasilkan oleh *regressor*.

Beberapa matriks yang dapat digunakan untuk evaluasi suatu *regressor* antara lain : *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), *Median Absolute Error*, score variansi dan dengan menghitung R^2 score (Joshi, Python Machine Learning Cookbook , 2016).

Pada penelitian ini nilai *Error* dievaluasi dengan penekanan pada *Mean Square Error* (MSE), *Mean Absolute Percent Error* (MAPE) dan R^2 score. MSE digunakan untuk menghitung tingkatan *squared error* pada prediksi dimana secara umum nilai *error* yang kecil menunjukkan performa *regressor* yang bagus. MSE dapat dihitung dengan menggunakan rumus (Santosa & Umam, 2018):

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (5)$$

dengan,

- y_i : nilai aktual
- \hat{y}_i : nilai prediksi dari y
- n : jumlah data observasi

R^2 score atau koefisien determinasi merupakan ukuran yang menunjukkan besar sumbangan dari variabel penjelas terhadap variabel respon. Rentang R^2 score adalah antara 0-1. Semakin dekat R^2 score ke nilai 1 maka regressor yang terbentuk dapat memprediksi variabel Y dengan baik (Siagian & Sugiarto, 2000). R^2 score dapat dihitung menggunakan rumus:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (6)$$

dengan,

- y_i : nilai aktual
- \hat{y}_i : nilai prediksi dari y
- \bar{y} : rata-rata nilai y

Mean Absolute Percent Error (MAPE) merupakan salah satu ukuran statistik yang digunakan untuk mengevaluasi keakuratan prediksi dalam bentuk persentase. MAPE banyak digunakan karena mudah ditafsirkan dan dijelaskan. Semakin rendah nilai MAPE, maka semakin baik model dalam meramalkan nilai. MAPE menghitung rata-rata dari keseluruhan persentase kesalahan antara nilai aktual dan nilai hasil prediksi dengan rumus:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| * 100\% \quad (7)$$

dengan,

- y_i : nilai aktual
- \hat{y}_i : nilai prediksi dari y
- N : jumlah data observasi

3.8 *Website Application* (flask)

Flask adalah *web framework* ringan yang ditulis dengan bahasa python. *Flask* dikategorikan dalam kerangka kerja mikro karena tidak memerlukan *tools* atau *library* tertentu dan memiliki *database* bawaan (Singh, Verma, & Parasher, 2019). *Flask* digunakan untuk membantu membuat *framework* sendiri dengan

menggunakan *toolkit* yang tersedia di python (DuPlain, 2013). Beberapa kelebihan *flask* adalah menyediakan *template* yang relatif sederhana untuk pengembangan aplikasi *web* dan memungkinkan untuk menulis aplikasi pada satu halaman yang sederhana, tetapi juga memiliki kekuatan untuk membangun aplikasi yang lebih besar tanpa masalah (Aggarwal, 2019).

Flask mulai muncul pada tahun 2010 berawal ketika Armin Ronacher yang sudah membangun *Werkzeug* (sebuah *web programming toolkit*) dan Jinja (*templating engine*) menggabungkan keduanya dan menambahkan *file* base64-encoded kedalam file python .py, kemudian menambahkan beberapa fungsi lain dan menamainya dengan *Denied microframework* (DuPlain, 2013). Meskipun termasuk *frameworks* pendatang baru dibanding dengan *Django* dan *Pylons*, *flask* dengan cepat mempunyai banyak peminat. *Flask* menyediakan *tools* yang kuat untuk pengembangan *web* yang lebih fleksibel dan memungkinkan para penggunanya untuk dapat menggunakan *library* mereka sendiri dan memilih setiap komponen yang terbaik untuk aplikasinya (Copperwaite & Leifer, 2015).

Flask memiliki tiga dependensi utama. Subsistem *routing*, *debugging*, dan *Web Server Gateway Interface* (WSGI) berasal dari *Werkzeug*, dukungan *template* disediakan oleh Jinja2, dan integrasi baris perintah berasal dari *Clik*. Semua dependensi ini ditulis oleh Armin Ronacher, pengembang Flask (Grinberg, 2018).

3.9 Label Encoder

Dalam pembelajaran *supervised* pada *machine learning*, jika data dalam bentuk angka maka algoritma dapat membaca atau mengoperasikanya secara langsung. Namun, sering kali label data dapat berupa bentuk kategorik ataupun bentuk teks, dimana harus diubah kedalam bentuk angka sehingga algoritma dapat menggunakannya langsung untuk memulai proses *training* data (Joshi, Python Machine Learning Cookbook , 2016).

Terdapat banyak cara untuk mengubah label kategorikal menjadi numerik, diantaranya adalah dengan menggunakan *Label-Encoder* yang merupakan salah satu fitur dari library SciKit-learn pada python. *Label Encoder* tersebut digunakan untuk mengubah teks atau data kategorikal menjadi data numerik dalam satu kolom data secara otomatis.

Berikut adalah contoh cara kerja label *encoder*:

Tabel 3.2 Contoh *Label Encoder*

Jenis kamar	Ketersediaan		Jenis kamar	Ketersediaan
Single room	2	Setelah proses label encoder →	0	2
Twin room	2		1	2
Double room	1		2	1
Family room	1		3	1
Twin room	3		1	3



BAB 4 METODOLOGI PENELITIAN

4.1 Populasi dan sampel

Populasi dari penelitian ini adalah data Airbnb diseluruh dunia, sedangkan sampel yang digunakan dalam penelitian adalah Airbnb Tokyo City yang dikumpulkan pada bulan November tahun 2019.

4.2 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini yaitu:

Tabel 4.1 Variabel Penelitian

Jenis Variabel	Nama Variabel
Variabel bebas atau variabel prediktor	<i>Neighbourhood</i> <i>Property Type</i> <i>Room Type</i> <i>Accommodates</i> <i>Bathrooms</i> <i>Bedrooms</i> <i>Beds</i> <i>Bed Type</i> <i>Cleaning Fee</i> <i>Guests Included</i> <i>Extra People</i> <i>Minimum Nights</i>
Variabel terikat atau variabel respon	<i>Price</i>

4.3 Definisi Operasional Variabel

Tabel 4.2 berikut berisi tentang definisi operasional dari masing-masing variabel yang digunakan pada penelitian:

Tabel 4.2 Definisi Operasional Variabel

Variabel	Definisi Operasional Variabel
<i>Neighbourhood</i>	Lingkungan lokasi tempat Airbnb berada
<i>Property Type</i>	Jenis akomodasi (bangunan)
<i>Room Type</i>	Jenis ruangan yang tersedia di Airbnb
<i>Accommodates</i>	Jumlah perorangan maksimal yang dapat ditampung dalam satu pesanan ruangan
<i>Bathrooms</i>	Jumlah kamar mandi yang tersedia di Airbnb
<i>Bedrooms</i>	Jumlah ruang tidur yang tersedia

Variabel	Definisi Operasional Variabel
<i>Beds</i>	Jumlah tempat tidur yang tersedia
<i>Bed Type</i>	Tipe tempat tidur yang tersedia
<i>Price</i>	Nilai harga sewa yang ditawarkan pada situs Airbnb
<i>Cleaning Fee</i>	Nilai biaya pelayanan kebersihan
<i>Guests Included</i>	Jumlah perorangan yang dapat penyewa terima sebagai tamu
<i>Extra People</i>	Nilai biaya tambahan orang ketiga
<i>Minimum Nights</i>	Jumlah minimal malam untuk dapat menyewa Airbnb

Nilai dari masing-masing variabel dalam penelitian didefinisikan dalam

Tabel 4.3 berikut ini:

Tabel 4.3 Nilai Variabel

No	Variabel	Nilai	Keterangan
1.	<i>Neighbourhood</i>	Adachi ku	Wilayah Adachiku
		Akiruno shi	Wilayah Akiruno shi
		Akishima Shi	Wilayah Akishima Shi
		Akishima Shi	Wilayah Akishima Shi
		Bunkyo Ku	Wilayah Bunkyo Ku
		Chiyoda Ku	Wilayah Chiyoda Ku
		Komae Shi	Wilayah Komae Shi
		Chuo Ku	Wilayah Chuo Ku
		Edogawa Ku	Wilayah Edogawa Ku
		Musashimurayama Shi	Wilayah Musashimurayama Shi
		Ome Shi	Wilayah Ome Shi
		Hachijo Machi	Wilayah Hachijo Machi
		Hachioji Shi	Wilayah Hachioji Shi
		Hamura Shi	Wilayah Hamura Shi
		Hinohara Shi	Wilayah Hinohara Shi
		Higashimurayama Shi	Wilayah Higashimurayama Shi
		Hino Shi	Wilayah Hino Shi
		Tama Shi	Wilayah Tama Shi
		Itabashi Shi	Wilayah Itabashi Shi
		Katsushika Ku	Wilayah Katsushika Ku
Arakawa Ku	Wilayah Arakawa Ku		
Kodaira Shi	Wilayah Kodaira Shi		
Koganei Shi	Wilayah Koganei Shi		
Chofu Shi	Wilayah Chofu Shi		
Kokubunji Shi	Wilayah Kokubunji Shi		

No	Variabel	Nilai	Keterangan
		Koto Ku	Wilayah Koto Ku
		Kunitachi Shi	Wilayah Kunitachi Shi
		Machida Shi	Wilayah Machida Shi
		Meguro Ku	Wilayah Meguro Ku
		Minato Ku	Wilayah Minato Ku
		Mitaka Shi	Wilayah Mitaka Shi
		Kuchu Shi	Wilayah Kuchu Shi
		Musashino Shi	Wilayah Musashino Shi
		Nakano Ku	Wilayah Nakano Ku
		Nerima Ku	Wilayah Nerima Ku
		Okutama Machi	Wilayah Okutama Machi
		Nishitokyo Shi	Wilayah Nishitokyo Shi
		Ogasawara Mura	Wilayah Ogasawara Mura
		Niijima Mura	Wilayah Niijima Mura
		Higashikurume Shi	Wilayah Higashikurume Shi
		Oshima Machi	Wilayah Oshima Machi
		Ota Ku	Wilayah Ota Ku
		Setagoya Ku	Wilayah Setagoya Ku
		Shibuya Ku	Wilayah Shibuya Ku
		Shinagawa Ku	Wilayah Shinagawa Ku
		Shinjuku Ku	Wilayah Shinjuku Ku
		Suginami Ku	Wilayah Suginami Ku
		Sumida Ku	Wilayah Sumida Ku
		Tachikawa Shi	Wilayah Tachikawa Shi
		Taito Ku	Wilayah Taito Ku
		Fussa Shi	Wilayah Fussa Shi
		Toshima Ku	Wilayah Toshima Ku
2.	<i>Property type</i>	<i>Aparthotel</i>	Hotel apartemen adalah kompleks apartemen yang menggunakan sistem pemesanan bergaya hotel. <i>Aparthotel</i> mirip dengan apartemen, tetapi tanpa kontrak tetap dan penghuni dapat <i>check out</i> kapan pun mereka mau.
		<i>Apartment</i>	<i>Apartment</i> yaitu sebuah tempat tinggal (terdiri atas kamar duduk, kamar tidur, kamar mandi, dapur, dan sebagainya) yang berada pada satu lantai bangunan bertingkat yang besar dan dilengkapi dengan berbagai fasilitas.

No	Variabel	Nilai	Keterangan
		<i>Boutique hotel</i>	<i>Boutique hotel</i> adalah hotel kecil yang bergaya serta modis dan bukan merupakan bagian dari jaringan hotel atau grup hotel dari sebuah perusahaan.
		<i>Tent</i>	Sebuah tenda yaitu tempat perlindungan portabel yang terbuat dari kain, ditopang oleh satu atau lebih tiang dan diregangkan dengan tali atau <i>loop</i> yang melekat pada pasak yang didorong ke tanah.
		<i>Cabin</i>	<i>Cabin</i> dapat diartikan sebagai tempat tinggal atau rumah kecil yang terbuat dari kayu dan terletak di daerah liar atau terpencil.
		<i>Camper/RV</i>	<i>Camper</i> adalah kendaraan bermotor besar dengan fasilitas untuk tidur dan memasak yang biasanya digunakan saat sedang berkemah.
		<i>Condominium</i>	<i>Condominium</i> berarti sebuah bangunan atau kompleks bangunan yang berisi sejumlah apartemen atau rumah yang dimiliki secara individual. Sistem kepemilikan kondominium tidak seperti apartemen yang disewa oleh penyewa, namun kondominium dimiliki secara langsung, di mana pemilik memiliki hak penuh untuk apartemen atau rumah individu tersebut.
		<i>Nature lodge</i>	<i>Nature lodge</i> adalah fasilitas akomodasi di kawasan wisata modern dengan konsep tetap menjaga kawasan natural, biasanya terletak di kawasan lingkungan alami di luar area taman langsung dan jauh dari pemukiman manusia.
		<i>Earth house</i>	Merupakan tempat tinggal dengan struktur bangunan

No	Variabel	Nilai	Keterangan
			berupa tanah yang menempel di dinding, di atap, atau yang sepenuhnya terkubur di bawah tanah. Sehingga membuatnya lebih mudah untuk mempertahankan suhu udara dalam ruangan yang stabil dan juga mengurangi biaya energi untuk pemanasan atau pendinginan.
		<i>Guesthouse</i>	<i>Guesthouse</i> merupakan bentuk jenis akomodasi berupa rumah biasa yang sebagian kamar-kamarnya disewakan kepada tamu. Perbedaan <i>Guesthouse</i> dengan hotel lainnya adalah tarifnya yang cenderung lebih murah dan <i>guesthouse</i> dikelola oleh pemiliknya sendiri yang biasanya tinggal di bangunan yang sama.
		<i>Hostel</i>	<i>Hostel</i> merupakan akomodasi penginapan dengan bentuk terbilang lebih sederhana. Hotel menyediakan ruangan tempat tidur pribadi atau berupa <i>dormitory</i> . Fasilitas yang tersedia di <i>hostel</i> biasanya digunakan bersama oleh tamunya. Seperti kamar mandi, ruang duduk, dapur, dan alat masak.
		<i>Hotel</i>	Hotel adalah suatu usaha yang menyediakan penginapan dalam satu bangunan yang disediakan secara khusus dan dapat disewakan dalam jangka pendek dengan menyediakan berbagai macam fasilitas yang tersedia di dalamnya.
		<i>Loft</i>	Konsep <i>loft</i> banyak digunakan di negara maju seperti Amerika dan Eropa. <i>Loft apartment</i> adalah konsep unit dari sebuah apartemen yang terdiri dari 2

No	Variabel	Nilai	Keterangan
			lantai, tapi tidak sepenuhnya 2 lantai. Karena biasanya luas lantai kedua tidak seluas lantai utama. Dalam penggunaan Inggris, <i>loft</i> biasanya cenderung menjadi kamar langsung di bawah atap yang diakses melalui tangga.
		<i>Dorm</i>	Bangunan yang menyediakan tempat tidur dan tempat tinggal untuk sejumlah orang banyak seperti sekolah asrama, sekolah menengah, perguruan tinggi atau mahasiswa.
		<i>Ryokan</i>	Merupakan penginapan dengan fasilitas dan bangunan yang berarsitektur Jepang. penginapan jenis ini menyediakan kamar bergaya Jepang yang berlantaikan tatami.
		<i>Serviced apartment</i>	Merupakan apartemen berperabotan lengkap yang disewakan untuk jangka pendek atau jangka panjang. <i>Serviced apartment</i> biasanya menawarkan fasilitas seperti hotel apartemen tetapi menyediakan lebih banyak ruang dan tersedia pelayanan seperti <i>house keeping</i> .
		<i>Bungalow</i>	Penginapan jenis ini biasanya berupa bangunan berbentuk rumah-rumah ataupun pondok-pondok yang bangunannya terpisah. bangunan bungalow biasanya memiliki arsitektur tradisional, dengan atap rendah, dengan beranda yang luas dan biasanya banyak dibangun di daerah pantai dan pegunungan.
		<i>Town house</i>	Merupakan kompleks hunian yang umumnya berlokasi di tengah kota dengan rumah berbentuk serupa dan Jumlah rumahnya terbatas, tidak sebanyak kompleks perumahan

No	Variabel	Nilai	Keterangan
			biasa. Biasanya juga dilengkapi dengan sistem keamanan satu akses.
		<i>Villa</i>	<i>Villa</i> adalah rumah hunian yang biasanya berbentuk rumah perorangan dan biasanya disewakan untuk tempat tinggal sekaligus berlibur. Untuk tipe penginapan <i>villa</i> ini harus menyewa keseluruhan bangunan. <i>Villa</i> biasanya terletak di daerah yang jauh dari keramaian, hawanya sejuk dan suasananya asri. Misalnya di daerah pegunungan atau pantai, maupun di dekat lokasi wisata.
3.	<i>Room type</i>	<i>Entire room</i>	Para tamu memiliki seluruh tempat untuk diri mereka sendiri, termasuk kamar tidur, kamar mandi, dan dapur.
		<i>Private room</i>	Para tamu memiliki kamar pribadi mereka sendiri untuk tidur sedangkan area lain dapat dibagikan.
		<i>Hotel room</i>	Para tamu berada di satu ruangan sendiri yang biasanya terdiri dari tempat tidur dan kamar mandi dan terletak bersebelahan dengan kamar hotel yang lain.
		<i>Shared room</i>	Para tamu tidur di kamar tidur atau area umum yang bisa dibagi dengan orang lain.
4.	<i>Accommodates</i>	1, 2, 3, ...	Jumlah perorangan maksimal yang dapat ditampung dalam satu pesanan ruangan
5.	<i>Bathrooms</i>	1, 2, 3, ..., 5	Jumlah kamar mandi
6.	<i>Bedrooms</i>	1,2,3,4	Banyaknya ruang atau kamar tidur yang tersedia.
7.	<i>Beds</i>	1,2,3, ...	Banyaknya kasur atau tempat tidur yang tersedia.
8.	<i>Bed type</i>	<i>Air bed</i>	Merupakan kasur udara atau kasur tiup yang terbuat dari <i>polyvinyl chloride</i> (PVC) atau <i>urethane</i> plastik tekstil

No	Variabel	Nilai	Keterangan
			diperkuat atau karet. <i>Airbed</i> dapat dipadatkan dan dibawa atau disimpan dalam bentuk kecil.
		<i>Futon</i>	<i>Futon</i> adalah jenis perangkat tidur tradisional Jepang yang digelar di atas tatami, di atas tempat tidur, atau kasur. Satu set <i>futon</i> terdiri dari shikibuton sebagai alas tidur dan kakebuton yang lebih lunak sebagai selimut.
		<i>Pull-out sofa</i>	Yaitu sofa yang dapat dijadikan sebagai tempat tidur dengan menurunkan engselnya yang dilapis kembali ke posisi horizontal atau dengan menarik keluar kasur yang tersembunyi.
		<i>Real bed</i>	<i>Real bed</i> adalah perabot yang digunakan sebagai tempat tidur atau bersantai. Sebagian besar tempat tidur modern terdiri dari kasur empuk dan diletakkan di atas alas yang kokoh, sering juga berupa bilah kayu.
9.	<i>Cleaning fee</i>	Misal: ¥6569, ¥4000, ¥5474	Berupa biaya yang dibayarkan untuk penambahan pelayanan kebersihan penginapan.
10.	<i>Guests Included</i>	1,2,3,4,5,6. ...	Banyaknya tamu yang dapat diterima oleh penyewa.
11.	<i>Extra People</i>	Misal: ¥3000, ¥5000, ¥1500, ¥2000	Berupa biaya yang dibayarkan untuk tamu tambahan.
12.	<i>Minimum Nights</i>	1,2,3, ...	Masa inap minimum yang dapat dipesan oleh penyewa.

4.4 Metode Pengumpulan Data

Penelitian ini menggunakan data sekunder yang diambil *website* resmi Airbnb yaitu <http://insideairbnb.com/get-the-data.html>. Data diunduh pada tanggal 8 Desember 2019 berupa informasi ringkasan Airbnb Tokyo sebanyak 13.726 data Airbnb yang berada di Tokyo.

4.5 Metode Penelitian

Metode yang digunakan pada penelitian ini adalah metode *Random Forest Regreesion* dimana *Random Forest* merupakan suatu metode bagian dari *decision tree* dan merupakan salah satu algoritma *machine learning*. Metode *Random Forest* ini digunakan untuk membuat model yang dapat memprediksi harga sewa Airbnb Tokyo. Kemudian model yang sudah didapatkan akan diterapkan kedalam *flask* atau *website application* dengan menggunakan *library flask*.

4.6 Perangkat Penelitian

Model didapatkan dengan melakukan proses *training (Machine learning)* sehingga membutuhkan perangkat *hardware* seperti laptop/komputer dengan spesifikasi yang cukup untuk mendapatkan suatu model yang dapat memprediksi data baru. Berikut ini adalah spesifikasi perangkat yang digunakan :

- Windows : Windows 10
- *Processor* : Intel® Core™ i3-7020U
- CPU : 2.30GHz 2.30GHz;
- RAM : 4GB.

Selanjutnya adalah beberapa *software* dan *library* yang di gunakan pada penelitian ini adalah:

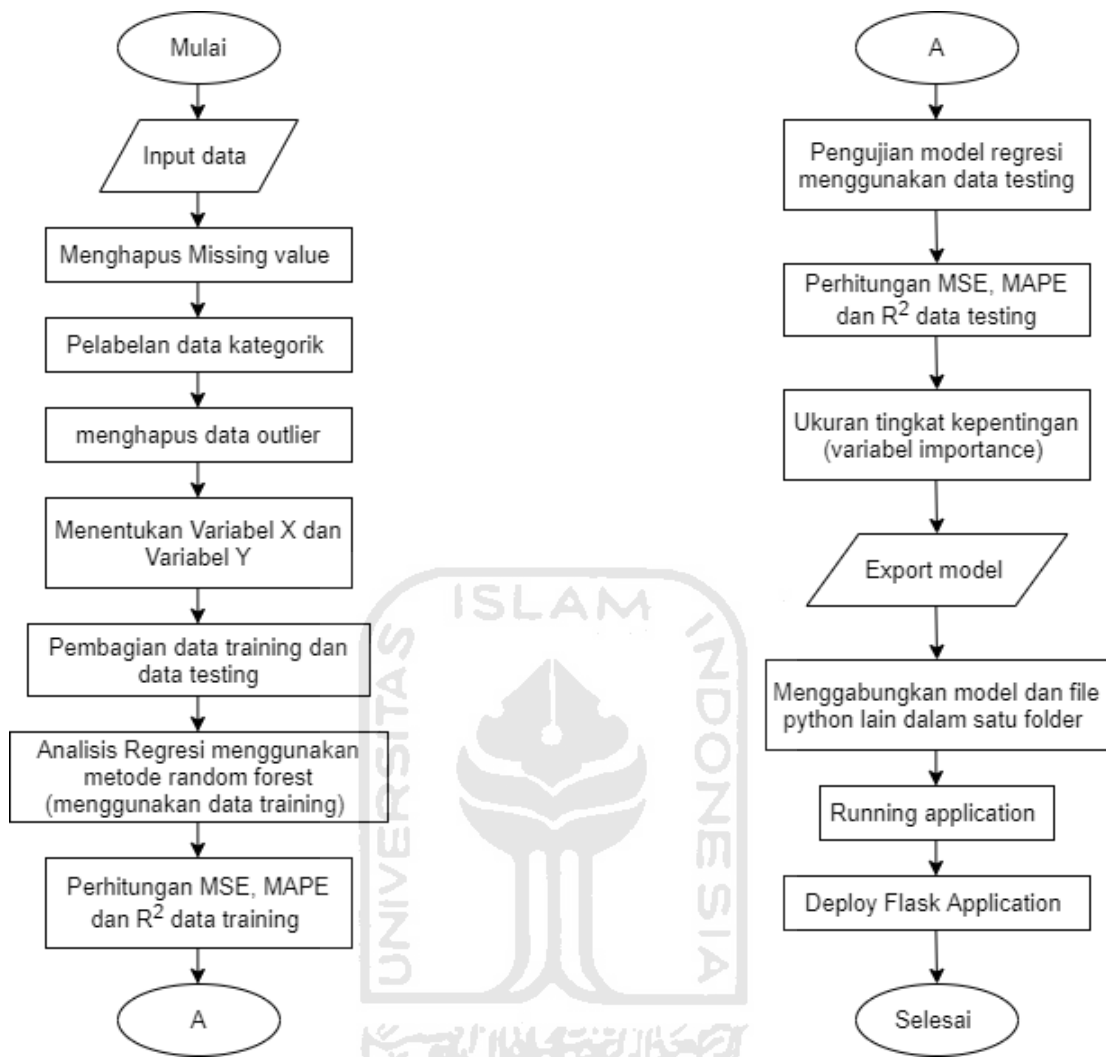
- Python
Sebuah aplikasi (*Software*) bahasa pemrograman yang digunakan untuk menjalankan *script* perintah.
- Numpy
Pustaka python untuk melakukan operasi matriks.
- Pandas
Pustaka python untuk melakukan analisis data atau olah data.
- Pickle
Pustaka python yang digunakan untuk menyimpan model atau menyimpan objek python kedalam *file*.
- Sklearn
Pustakan python yang digunakan untuk *processing* data atau *training* data.
- Flask

Merupakan salah satu *micro web framework* dari bahasa python.

4.7 Tahapan Penelitian

Pada penelitian ini adapun tahap-tahap yang dilakukan adalah sebagai berikut :

1. Melakukan pemeriksaan *missing value* pada data.
2. Melakukan pelabelan pada dataset yang berkategori *string* atau *Boolean* menggunakan *label encoder*.
3. Melakukan pemeriksaan data *outlier*, kemudian menghapus data *outlier*.
4. Menentukan variabel yang akan dijadikan variabel *dependen* dan variabel *independen*.
5. Menentukan jumlah data *training* dan data *testing* yang akan digunakan.
6. Regresi data dengan metode *Random Forest* menggunakan data *training* yang sudah ditentukan sebelumnya.
7. Menghitung nilai MSE dan R^2 yang didapatkan dari hasil uji data *training*.
8. Menguji model *Random Forest* yang dihasilkan menggunakan data *testing*.
9. Menghitung nilai MSE dan R^2 yang didapatkan dari hasil uji data *testing*.
10. Membuat *plot* ukuran tingkat kepentingan (*variable importance*) dari variabel *independent* (variabel prediktor).
11. Melakukan *export* model hasil *training* metode *Random Forest* dengan ekstensi.pkl, dimana format .pkl merupakan suatu format yang dapat digunakan untuk menyimpan bentuk fungsi atau model kedalam suatu bentuk file.
12. Menggabungkan satu folder export model *Random Forest*, dataset dalam format .csv, dan *css template website* yang sudah disusun sebelumnya.
13. *Running application* melalui *anaconda prompt* atau *command prompt*.
14. Setelah program berhasil, selanjutnya *deploy project application* ke Heroku agar *website* dapat dilihat oleh publik.



Gambar 4.1 Alur Penelitian

BAB 5 HASIL DAN PEMBAHASAN

5.1 *Preprocessing Data*

Data yang tersedia perlu dilakukan pembersihan data atau *preprocessing* data menjadi format yang tepat. Tujuannya agar data siap untuk menjadi *training* dan konsumsi algoritma *machine learning* sehingga dapat menghasilkan kualitas hasil *machine learning* yang lebih baik. Proses pembersihan data meliputi diantaranya mengidentifikasi dan me-*remove* data yang mempunyai *missing value*, *outlier* dan melakukan pelabelan data kategorik menjadi data numerik.

5.1.1 *Label Encoder*

Dalam melakukan klasifikasi atau regresi, terkadang akan berurusan dengan data yang berbentuk kata-kata atau sesuatu yang lain. Sehingga untuk melakukan tahap *training* diperlukan pelabelan masing-masing kategori kedalam bentuk numerik. Untuk mengonversi label kata menjadi angka, peneliti menggunakan pembuat label *encoding* yang merupakan salah satu fitur dari *library* SciKit-learn pada python.

Pada penelitian ini variabel yang akan dikonversi yaitu:

- *Property type*

Tabel 5.1 Konversi Variabel *Property type*

Data asli	Hasil konversi
<i>Aparthotel</i>	0
<i>Apartment</i>	1
⋮	⋮
<i>Villa</i>	24

- *Room type*

Tabel 5.2 Konversi Variabel *Room Type*

Data asli	Hasil konversi
<i>Entire room</i>	0
<i>Private room</i>	1
<i>Hotel room</i>	2
<i>Shared room</i>	3

- *Neighbourhood*

Tabel 5.3 Konversi Variabel *Neighbourhood*

Data asli	Hasil konversi
Adachi ku	0
Akiruno shi	1
Koganei shi	2
Kita ku	3
Bunkyo Ku	4
Chiyoda Ku	5
⋮	⋮
Toshima ku	51

- *Bed type*

Tabel 5.4 Konversi Variabel *Bed Type*

Data asli	Hasil konversi
<i>Air bed</i>	0
<i>Futon</i>	1
<i>Pull-out sofa</i>	2
<i>Real bed</i>	3

5.1.2 Penentuan Data *Training* dan Data *Testing*

Setelah data melalui tahap *preprocessing* dengan menghilangkan *missing value*, menghapus data *outlier* dan mengubah data kategorik menjadi data numerik menggunakan *label encoder*, selanjutnya data dibagi menjadi data *training* dan data *testing*. Data *training* digunakan untuk membuat model *machine learning*, sedangkan data *testing* digunakan untuk menguji performa *output* model yang dihasilkan.

Peneliti menggunakan persentase data *training* dan data *testing* sebagai berikut:

Tabel 5. 5 Persentase Pembagian Data *Training* dan Data *Testing*

	<i>Training</i>	<i>Testing</i>	Total
Persentase	70%	30%	100%
Data yang digunakan	6274	2689	8962

5.2 *Random Forest Regression*

5.2.1 *Prediksi dan Validasi Model*

Model *Random Forest regressor* terbentuk berdasarkan *training* data sampel yang telah ditentukan sebelumnya. Variabel *price* menjadi variabel dependen dengan tipe data numerik yang akan diprediksi nilainya berdasarkan variabel *Neighbourhood, Property Type, Room Type, Accommodates, Bathrooms, Bedrooms, Beds, Bed Type, Cleaning Fee, Guests Included, Extra People, Minimum Nights*.

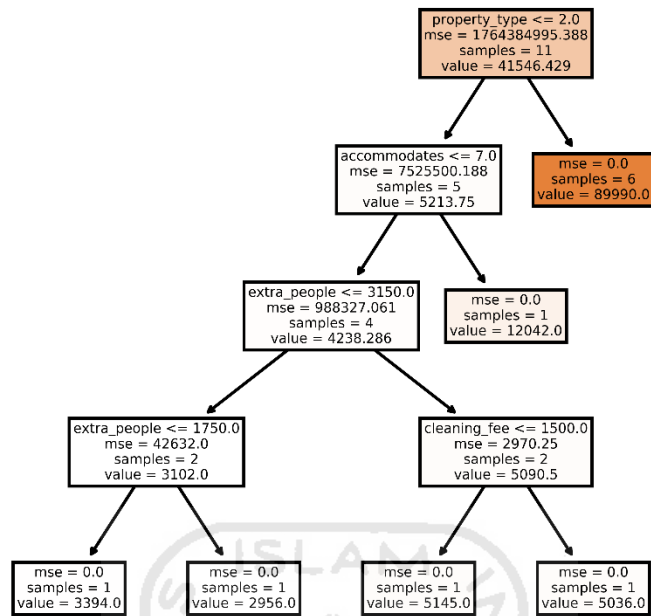
Pembentukan model *Random Forest regressor* dimulai dengan membangun hutan pohon. Penumbuhan pohon yang tepat akan menghasilkan hasil prediksi yang lebih akurat. Pemilihan jumlah penumbuhan pohon dilakukan dengan membandingkan beberapa nilai eror yang dihasilkan dengan jumlah pohon yang berbeda-beda. Jumlah pohon dengan nilai eror terkecil akan dipilih untuk digunakan dalam pembentukan model *regressor*.

Tabel 5.6 Penentuan Jumlah Pohon

Jumlah Pohon	MAPE (%)
50	24%
100	23.82%
200	23.67%
250	23.7%
300	23.76

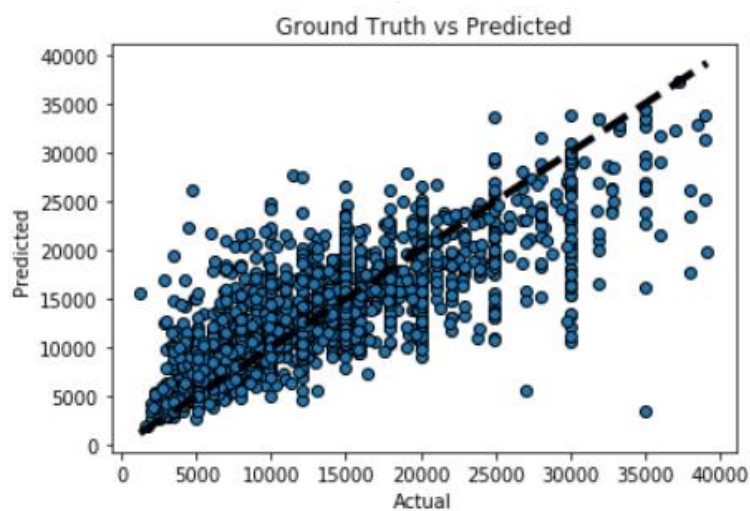
Berdasarkan **tabel 5.6** diatas, jumlah pohon yang menghasilkan nilai eror terkecil adalah sejumlah 200 pohon dengan nilai eror sebesar 23.67%. Dengan demikian jumlah *n_estimators* yang digunakan dalam penelitian ini adalah sebanyak 200.

Berikut adalah contoh *output* pohon *Random Forest* dengan menggunakan variabel *Property Type, Accommodates, Beds, Cleaning Fee, Extra People* dari dataset Airbnb bagian distrik Hachioji Shi:



Gambar 5.1 Contoh *Output* Pohon *Random Forest Regression*

Tujuan dari sebuah regresi adalah untuk dapat memprediksi respon dari model yang telah dihasilkan. Salah satu kriteria model regresi yang baik adalah ketika data hasil prediksi mendekati dengan data aktualnya. Model *regressor* yang telah dibangun menggunakan data training dengan $n_estimators$ sebanyak 200 akan diperoleh grafik data aktual dengan data hasil prediksi yang dihasilkan model *regressor* menggunakan *Random Forest* seperti pada gambar berikut:



Gambar 5.2 Grafik Data Asli dengan Data Hasil Prediksi

Garis *trend* warna hitam adalah garis $y = x$ dengan y adalah nilai prediksi, sedangkan x adalah nilai aktual. Apabila nilai prediksi mendekati atau sama dengan nilai aktual, maka titik data berada tepat atau dekat dengan garis *trend*. Berdasarkan **gambar 5.1**, ketika nilai x, y diantara 0 – 25,000 titik-titik data berada pada garis trend dan juga banyak mendekati garis *trend*. Namun, ketika nilai x, y di atas 25,000; titik-titik data jarang yang berada atau mendekati garis *trend*. Semakin titik data berada mendekati garis *trend*, berarti nilai hasil prediksi semakin mendekati nilai aktual data.

Tabel 5.7 Nilai Prediksi dan Nilai Aktual (Yen)

No	Hasil data asli	Harga prediksi
1	14,451.0	15,908.95
2	16,969.0	14,725.10
3	13,028.0	14,683.47
4	13,466.0	14,022.61
5	9,962.0	13,326.16
6	4,817.0	5,596.53
7	20,034.0	24,896.12
8	9,943.0	7,498.71
9	14,998.0	15,033.58
10	20,034.0	19,720.16
11	9,962.0	10,687.74
12	11,386.0	11,620.92
13	9,962.0	13,487.27
14	6,569.0	7,700.08
15	5,036.0	7,300.52
16	6,021.0	5,990.23
17	31,967.0	33,511.14
18	20,034.0	18,887.21
19	14,013.0	11,958.52
20	14,998.0	15,555.33
21	6,897.0	6,637.22
22	3,503.0	2,923.27
23	7,992.0	9,374.33
24	14,998.0	13,808.83
⋮	⋮	⋮
	20,034.0	24,896.12

Tabel 5.7 diatas merupakan perbandingan nilai data *testing* hasil prediksi dengan data aktual harga sewa Airbnb. Dari tabel tersebut dapat dilihat bahwa antara harga hasil prediksi dan harga asli terdapat beberapa prediksi yang memiliki nilai yang berbeda dengan harga asli. Seperti contoh data ke-16 dimana prediksi harga sewa Airbnb yang dihasilkan adalah sebesar ¥13,487.27 sedangkan harga asli sewa Airbnb tersebut adalah sebesar ¥9,962. Namun, tidak sedikit pula nilai prediksi yang dihasilkan tidak berbeda jauh seperti pada data ke-25 dengan hasil prediksi sewa sebesar ¥6,637.22 sedangkan harga asli sewa sebesar ¥6,897.

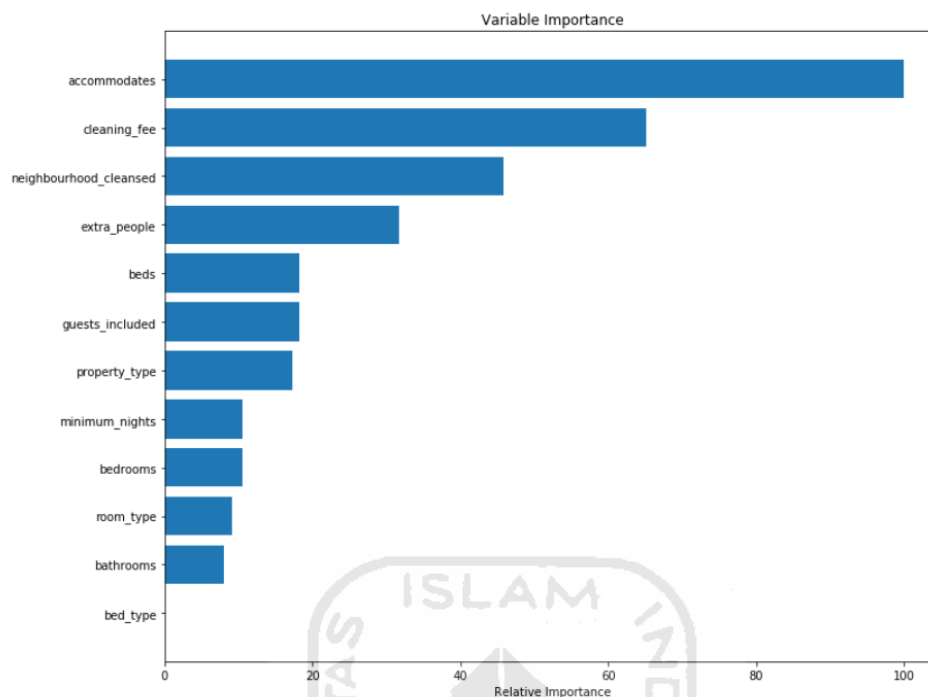
Tabel 5.8 Hasil Akurasi

MSE	MAPE	R ²
15,389,924.24	23.76%	0.68 atau 68%

Pembentukan model pada *Random Forest regressor* dengan menggunakan 200 pohon mendapatkan nilai MSE sebesar 15,389,924.24. Nilai MSE ini menggambarkan tingkatan eror hasil prediksi performa *regressor* secara umum. Nilai eror yang kecil menunjukkan performa *regressor* yang bagus. Kemudian nilai MAPE yang dihasilkan sebesar 23.76%. Nilai MAPE ini menunjukkan rata-rata dari keseluruhan persentase kesalahan hasil prediksi antara nilai aktual dan nilai hasil prediksi. Seperti halnya MSE, pada nilai prediksi MAPE semakin kecil nilai MAPE maka tingkat kesalahan prediksi juga semakin kecil. Selanjutnya nilai yang menunjukkan besarnya sumbangan dari variabel prediktor terhadap variabel respon yaitu nilai R². Pada penelitian ini R² *score* yang dihasilkan yaitu sebesar 68% yang berarti bahwa variabel prediktor yang diinputkan berpengaruh terhadap variabel respon (*price*) sebesar 68%.

5.2.2 Variabel Importance

Salah satu *output* yang dihasilkan dalam analisis *Random Forest Regreesion* adalah variabel *importance*, yaitu menunjukkan ukuran kepentingan setiap variabel respon atau variabel prediktor dengan variabel lainnya. Dalam *output* ini juga dapat digunakan untuk mengidentifikasi prediktor yang relevan dari sejumlah variabel kandidat.



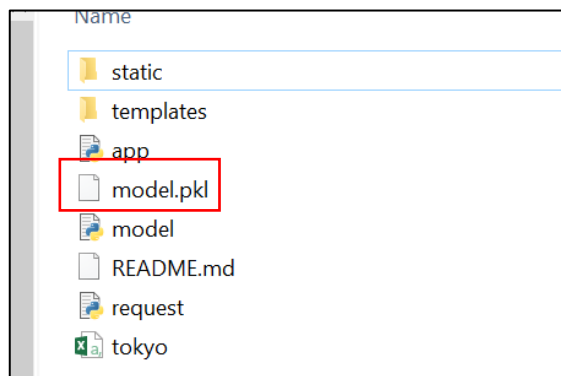
Gambar 5.3 *Output Variabel Importance*

Dari *output* pada **gambar 5.3** diatas dapat diketahui bahwa variabel yang paling berpengaruh dalam penentuan prediksi harga sewa Airbnb adalah variabel ‘*accommodates*’ dibandingkan dengan 11 variabel lain yang di-*input*-kan. *Accommodates* adalah jumlah perorangan dalam satu pesanan ruangan pada Airbnb. Sementara variabel ‘*bed_type*’ tidak berpengaruh dalam prediksi harga Airbnb. Urutan kepentingan variabel setelah variabel *accommodates* yaitu variabel *cleaning_fee*, *neighbourhood_cleansed*, *extra_people*, *guest_included*, *beds*, *property_type*, *minimum_nights*, *bedrooms*, *room_type*, kemudian *bathrooms*. Urutan nilai kepentingan variabel ini dapat menunjukkan variabel prediktor mana yang paling relevan dalam memprediksi harga sewa Airbnb.

5.3 Membuat *Web Application* Menggunakan Flask

5.3.1 *Export Model Random Forest*

Langkah awal dalam membuat situs *web* adalah dengan meng-*export* model *machine learning* yang sudah dibuat sebelumnya. *Export* model dilakukan dengan menggunakan *library* ‘*pickle*’ pada python. Model akan tersimpan pada direktori python peneliti dan memiliki format ‘*model.pkl*’.



Gambar 5.4 *Export Model Random Forest*

5.3.2 Import Package Flask

Selanjutnya *import package flask* dengan menuliskan *script* berikut pada notepad++ dan menyimpannya dengan format 'app.py' kemudian disimpan dalam satu file yang sama dengan file 'model.pkl'.

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Rental
Fees are approaching $ {}'.format(output))

@app.route('/predict_api', methods=['POST'])
def predict_api():
    '''
    For direct API calls through request
    '''
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
```


5.3.4 Menampilkan Halaman *Website*

Untuk dapat menampilkan halaman *website*, pindahkan direktori kedalam satu *file* yang menyimpan model *machine learning*, *index.html* dan *app.py*. Dalam hal ini peneliti menyimpannya dengan nama folder Airbnb.

```

Anaconda Prompt (Anaconda3)

(base) C:\Users\Acer>cd D:\AIRBNB

(base) C:\Users\Acer>D:

(base) D:\AIRBNB>

```

Gambar 5.5 Direktori Folder Airbnb

Selanjutnya ketikkan perintah `python app.py` sehingga akan muncul alamat IP yang digunakan untuk masuk kedalam *website*.

```

(base) D:\AIRBNB>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 264-070-112
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Gambar 5.6 Menampilkan Alamat IP Website

Ketikkan alamat IP yang sudah didapat kedalam *chrome* untuk membuka *website*. Berikut tampilan *website* prediksi harga Airbnb:

The screenshot shows a web browser window titled 'Airbnb Price Prediction' with the address '127.0.0.1:5000'. The website interface is blue and white. It features a form with the following fields:

- Neighborhood: dropdown menu (selected: Adachi-ku)
- Property Type: dropdown menu (selected: Apartment)
- Room Type: dropdown menu (selected: Entire room)
- Accommodates: text input (placeholder: Enter How Many People Will Stay)
- Bathrooms: text input (placeholder: Enter The Number of Bathrooms)
- Bedrooms: text input (placeholder: Enter The Number of Bedrooms)
- Beds: text input (placeholder: Enter The Number of Beds)
- Bed Type: dropdown menu (selected: All bed)
- Cleaning fee: text input (placeholder: Enter Additional Cleaning Fee in Yen)
- Security Included: text input (placeholder: Enter How Many Guest)
- Extra People: text input (placeholder: Enter Additional Fees for Extra People)
- Minimum Nights: text input (placeholder: Enter Minimum Nights Will Stay)

At the bottom right of the form, there is a yellow 'Check Price' button. An 'Activate Windows' watermark is visible in the bottom right corner of the browser window.

Gambar 5.7 Tampilan *Website* Prediksi Harga Airbnb

5.3.5 Prediksi Harga Airbnb

Setelah *website* dapat diakses selanjutnya mencoba *website* untuk memprediksi harga Airbnb.

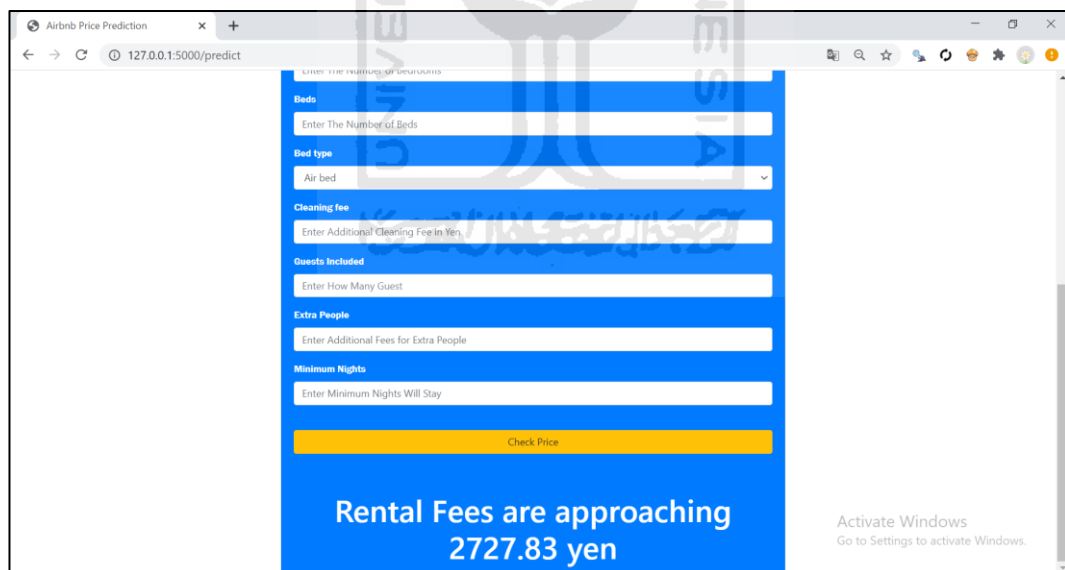


The screenshot shows a web browser window titled "Airbnb Price Prediction" with the URL "127.0.0.1:5000". The page features a blue-themed input form with the following fields:

- Neighborhood: Dropdown menu (value: "Ede Kru")
- Property Type: Dropdown menu (value: "Apartment")
- Room type: Dropdown menu (value: "Entire room")
- Accommodates: Text input (value: "2")
- Bathrooms: Text input (value: "1")
- Cookbooks: Text input (value: "1")
- Beds: Text input (value: "2")
- Bed type: Dropdown menu (value: "Air bed")
- Cleaning fee: Text input (value: "1000")
- Guests included: Text input (value: "1")
- Extra People: Text input (value: "2000")
- Minimum Nights: Text input (value: "1")

At the bottom of the form is a yellow "Check Price" button. A watermark for "UNIVERSITAS ISLAM INDONESIA" is visible in the background.

Gambar 5.8 *Input* Nilai Variabel Prediktor



The screenshot shows the same web browser window, but the form is now filled out and the "Check Price" button has been clicked. The output is displayed in a large blue box:

Rental Fees are approaching
2727.83 yen

The input fields are now empty. A watermark for "UNIVERSITAS ISLAM INDONESIA" is visible in the background.

Gambar 5.9 *Output* Variabel Respon

Dari **gambar 5.9** diatas dapat diketahui bahwa model *machine learning* sudah berhasil dibuat dalam *website*. Selanjutnya *web app* tersebut dapat di *deploy* menggunakan Heroku untuk dapat digunakan secara publik.

BAB 6 PENUTUP

6.1. Kesimpulan

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Harga sewa Airbnb diprediksi dengan variabel bebas yaitu *Neighbourhood, Property Type, Room Type, Accommodates, Bathrooms, Bedrooms, Beds, Bed Type, Cleaning Fee, Guests Included, Extra People, Minimum Nights*. Berdasarkan analisis data dengan menggunakan metode *Random Forest regressor* diperoleh jumlah pohon atau *n_estimator* sebanyak 200. Pada *output* variabel *importance* diperoleh variabel '*Accommodates*' merupakan variabel yang paling berpengaruh dalam prediksi harga Airbnb. Variabel ini menunjukkan jumlah perorangan dalam satu pesanan ruangan pada Airbnb.
2. Tingkat akurasi hasil prediksi harga sewa Airbnb diukur dengan menggunakan nilai MSE, MAPE dan R^2 dari data *testing*. Nilai MSE didapatkan sebesar 15389924.243, Nilai MAPE didapatkan sebesar 23.67%, sedangkan nilai R^2 didapatkan sebesar 0.68 artinya variabel prediktor yang diinputkan berpengaruh terhadap variabel respon (*price*) sebesar 68%.
3. Situs *website* prediksi harga sewa Airbnb dibangun dengan memanfaatkan *webhosting* gratis dari Heroku. Model *Random Forest* diekspor dalam format *.pkl* dengan bantuan *flask* kemudian digabungkan dengan format desain halaman *website* yang sudah ditentukan sebelumnya. Hasilnya adalah situs web dengan URL <https://airbnb-tokyo-price-prediction.herokuapp.com/>

6.2. Saran

Dari penelitian yang telah dilakukan, maka peneliti dapat memberikan saran diantaranya sebagai berikut :

1. Pada penelitian selanjutnya dapat digunakan variabel yang sesuai dengan *output* variabel *importance* atau dengan 5 *output* variabel *importance* yang teratas.

2. Meningkatkan *output* nilai R^2 dengan cara menambah jumlah data.
3. Pada bagian *missing value* data tidak dihapus atau dihilangkan, namun dapat diisi dengan suatu nilai yang diperoleh dari *modus* jika variabel tersebut merupakan data kategorik maupun nilai *mean* data jika variabel tersebut merupakan data numerik sehingga bisa menambah jumlah data.
4. Desain halaman *website* yang lebih kreatif dan komunikatif dibutuhkan untuk pengembangan *website* yang lebih baik.



DAFTAR PUSTAKA

- Aggarwal, S. (2019). *Flask Framework Cookbook: Over 80 Proven Recipes And Techniques For Python Web Development With Flask, 2nd Edition*. UK: Packt Publishing.
- Annisa, R. (2019). Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Penderita Penyakit Jantung. *Jurnal Teknik Informatika Kaputama (JTIK)* , 22.
- Aryo, Y., Padmadisastra, S., & Chadidjah, A. (2018). Analisis Perbandingan Kinerja Cart Konvensional, Bagging dan *Random Forest* Pada Klasifikasi Objek: Hasil Dari Dua Simulasi. *Media Statistika*.
- Budi A, I. M. (2015). Prediksi Lama Studi Mahasiswa Dengan Metode *Random Forest* (Studi Kasus : Stikom Bali). *Csrid Journal, Vol.8 No.3*, 201-208.
- Copperwaite, M., & Leifer, C. (2015). *Learning Flask Framework*. UK: Packt Publishing.
- Databoks.Katadata.Co.Id. (2019, 11 21). *Kota Paling Banyak Dikunjungi Turis Asing*. Retrieved From Databoks: <https://Databoks.Katadata.Co.Id/Datapublish/2019/11/21/20-Kota-Paling-Banyak-Dikunjungi-Turis-Asing#>
- Duplain, R. (2013). *Instant Flask Web Development*. UK: Packt Publishing Ltd.
- Gilles Louppe, L. W. (2013). Understanding Variable Importances In Forests Of Randomized Trees.
- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications With Python*. United States Of America: O'Reilly Media.
- Insideairbnb.Com. (2019). *Get The Data*. Retrieved From [Http://Insideairbnb.Com/Get-The-Data.Html](http://Insideairbnb.Com/Get-The-Data.Html)
- J. Lewis, R. (2000). An Introduction To Classification And Regression Tree (CART) Analysis. *Annual Meeting Of The Society For Academic Emergency Medicine*.

- Jatmiko, Y. A., Padmadisastra, S., & Chadidjah, A. (2019). Analisis Perbandingan Kinerja Cart Konvensional, Bagging Dan *Random Forest* Pada Klasifikasi Objek: Hasil Dari Dua Simulasi. *Media Statistika 12(2) 2019: 1-12*.
- Joshi, P. (2016). *Python Machine Learning Cookbook*. UK: Packt Publishing.
- Joshi, P. (2017). *Artificial Intelligence With Python*. UK: Packt Publishing.
- Liaw, A., & Wiener, M. (2001). Classification And Regression By Randomforest. 18.
- Lifestyle.Okezone.Com. (2018, September 19). *7 Makna Tak Terduga Logo Yang Sering Ditemui Ketika Bepergian*. Retrieved From <https://Lifestyle.Okezone.Com/Read/2018/09/19/406/1952637/7-Makna-Tak-Terduga-Logo-Yang-Sering-Ditemui-Ketika-Bepergian>
- Markey.Id. (2019). *Airbnb Adalah? Layanan Pemesanan Penginapan Online*. Retrieved From <https://Markey.Id/Blog/Bisnis/Airbnb-Adalah>
- Prasetya P, N., Lasama, J., Pradika E.P, A., & Prasetiadi, A. (2019). Memprediksi Ketinggian Tsunami Menggunakan *Random Forest* Regressor. *Conference On Electrical Engineering, Telematics, Industrial Technology, And Creativemedia 2019*, 48.
- Primajaya, A., & Sari, B. N. (2018). *Random Forest* Algorithm For Prediction Of Precipitation. *Indonesian Journal Of Artificial Intelligence And Data Mining (IJAIDM)*, 27-31.
- Riski, D. T. (2015). Keputusan Konsumen Dalam Memilih Hotel Pangeran Pekanbaru Riau. *Jom Fisip Vol 2 No. 2*.
- Santosa, B., & Umam, A. (2018). *Data Mining Dan Big Data Analytics : Teori Dan Implementasi Menggunakan Python & Apache Spark*. Yogyakarta: Penebar Media Pustaka.
- Sayad, S. (2010). *Decision Tree - Regression*. Retrieved From https://Www.Saedsayad.Com/Decision_Tree_Reg.Html
- Siagian, D., & Sugiarto. (2000). *Metode Statistika Untuk Bisnis Dan Ekonomi*. Jakarta: Gramedia Pustaka Utama.

- Singh, M., Verma, A., & Parasher, A. (2019). *Implementation Of Database Using Python Flask Framework. International Journal Of Engineering And Computer Science*, 24894-24899.
- Strobl, C., Boulesteix, A. L., Zeileis, A., & Hothorn, T. (2007). *Bias In Random Forest Variable Importance Measures: Illustrations, Sources And A Solution . BMC Bioinformatics*.
- Sumartinidan, S. H., & Purnami, S. W. (2015). *Penggunaan Metode Classification And Regression Trees (CART) Untuk Klasifikasi Rekurensi Pasien Kanker Serviks Di RSUD Dr. Soetomo Surabaya. Jurnal Sains Dan Seni ITS Vol. 4,*
- Tang, E., & Sangani, K. (2015). *Neighborhood And Price Prediction For San Francisco Airbnb Listings .*
- Techcrunch.Com. (2014, September 19). *Airbnb CFO Andrew Swain Has Left The Company*. Retrieved From <https://Techcrunch.Com/2014/09/18/Airbnb-Cfo-Andrew-Swain-Has-Left-The-Company/>
- Wibowo, A. (2016). *Analisis Perbandingan Kinerja Metode Klasifikasi Dalam Data Mining*.
- Worldpopulationreview.Com. (2020). *Tokyo Population 2020*. Retrieved From <https://Worldpopulationreview.Com/World-Cities/Tokyo-Population>

LAMPIRAN

Lampiran 1. *Script* training dan ekspor model

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
import warnings
import sklearn

data = pd.read_csv('tokyo.csv')

# Creating DV and IV sets
X = data.drop('price', axis=1)
y = data['price']

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.3, random_state=1234)

from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 200, random_state
= 1234)

#Fitting model with training data
regressor.fit(X_train, y_train)

# Saving model to disk
pickle.dump(regressor, open('model.pkl', 'wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl', 'rb'))
print(model.predict([[45, 14, 2, 4, 1, 1, 3, 3, 5474, 3, 0, 1]]))

#Variabel Importance
feature_importance = regressor.feature_importances_
feature_importance = 100.0 * (feature_importance /
feature_importance.max())
sorted_idx = np.argsort(feature_importance)
pos = np.arange(sorted_idx.shape[0]) + .5
plt.figure(figsize=(12,10))
plt.barh(pos, feature_importance[sorted_idx], align='center')
plt.yticks(pos, X_train.columns[sorted_idx])
plt.xlabel('Relative Importance')
plt.title('Variable Importance')
plt.show()

#Actual value vs predicted value
from sklearn.model_selection import cross_val_predict
fig, ax = plt.subplots()
ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
```

```
ax.plot([y_test.min(), y_test.max()], [y_test.min(),
y_test.max()], 'k--', lw=4)
ax.set_xlabel('Actual')
ax.set_ylabel('Predicted')
ax.set_title("Ground Truth vs Predicted")
plt.show()

predicts = regressor.predict(X_test)
error_airbnb = pd.DataFrame({
    'Actual Values': np.array(y_test).flatten(),
    'Predicted Values': predicts.flatten()})
error_airbnb.head(30)
```



Lampiran 2. Script App.py

```

import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    """
    For rendering results on HTML GUI
    """
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Rental
Fees are approaching {} yen'.format(output))

@app.route('/predict_api',methods=['POST'])
def predict_api():
    """
    For direct API calls through request
    """
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

if __name__ == "__main__":
    app.run(debug=True)

```

Lampiran 3. *Script halaman website*

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Airbnb Price Prediction</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/boots
trap.min.css" integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYxXfFc+NcPb1dKGj7Sk"
crossorigin="anonymous">
  <style>
    *{
      padding: 0px;
      margin: 0px;
    }
    body {
      background-image: #FFB6C1;
    }
    .card-header{
      text-align: center;
    }
    button{
      padding: 15px;
      margin: 40px 0;
    }
    h1{
      font: bold sans-serif ;
    }
    label{
      font-family: 'Franklin Gothic Medium', 'Arial Narrow',
Arial, sans-serif;
    }
    .result{
      font-size: 50px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="offset-sm-3 col-sm-6">
    <div class="card text-white bg-primary">
      <div class="card-header">
        <h1>Airbnb Price Prediction</h1>
      </div>

```

```

    <div class="card-body">
      <form action="{{ url_for('predict')}}"
method="post">

          <div class="form-group">
            <label>Neighbourhood</label>
            <select name="neighbourhood_cleansed"
id="neighbourhood_cleansed" class="form-control"
required="required">
              <option value="0">Adachi Ku</option>
              <option
value="1">Akiruno Shi</option>
              <option
value="2">Akisihma Shi</option>
              <option value="3">Kita
Ku</option>
              <option
value="4">Bunkyo Ku</option>
              <option
value="5">Chiyoda Ku</option>
              <option
value="6">Komae Shi</option>
              <option value="7">Chuo
Ku</option>
              <option
value="8">Edogawa Ku</option>
              <option
value="9">Musashimurayama Shi</option>
              <option value="10">Ome
Shi</option>
              <option
value="11">Hachijo Machi</option>
              <option
value="12">Hachioji Shi</option>
              <option
value="13">Hamura Shi</option>
              <option
value="14">Hinohara Shi</option>
              <option
value="15">Higashimurayama Shi</option>
              <option
value="16">Hino Shi</option>
              <option
value="17">Tama Shi</option>
              <option
value="18">Itabashi Shi</option>
              <option
value="19">Katsushika Ku</option>
              <option
value="20">Arakawa Ku</option>
              <option
value="21">Kodaira Shi</option>
              <option
value="22">Koganei Shi</option>

```

value="23">Chofu Shi</option> <option
 value="24">Kokubunji Shi</option> <option
 value="25">Koto Ku</option> <option
 value="26">Kunitachi Shi</option> <option
 value="27">Machida Shi</option> <option
 value="28">Meguro Ku</option> <option
 value="29">Minato Ku</option> <option
 value="30">Mitaka Shi</option> <option
 value="31">Kuchu Shi</option> <option
 value="32">Musashino Shi</option> <option
 value="33">Nakano Ku</option> <option
 value="34">Nerima Ku</option> <option
 value="35">Okutama Machi</option> <option
 value="36">Nishitokyo Shi</option> <option
 value="37">Ogasawara Mura</option> <option
 value="38">Niijima Mura</option> <option
 value="39">Higashikurume Shi</option> <option
 value="40">Oshima Machi</option> <option
 Ku</option> <option value="41">Ota
 value="42">Setagoya Ku</option> <option
 value="43">Shibuya Ku</option> <option
 value="44">Shinagawa Ku</option> <option
 value="45">Shinjuku Ku</option> <option
 value="46">Suginami Ku</option> <option
 value="47">Sumida Ku</option> <option
 value="48">Tachikawa Shi</option> <option
 value="49">Taito Ku</option> <option
 value="50">Fussa Shi</option> <option


```

value="51">Toshima Ku</option>
</div>
</select>
<div class="form-group">
  <label>Property Type</label>
  <select name="property_type"
id="property_type" class="form-control" required="required">
    <option
value="0">Aparthotel</option>
    <option
value="1">Apartment</option>
    <option value="2">Bed
and Breakfast</option>
    <option
value="3">Boutique hotel</option>
    <option
value="4">Tent</option>
    <option
value="5">Cabin</option>
    <option
value="6">Camper/RV</option>
    <option
value="7">Condominium</option>
    <option
value="8">Nature lodge</option>
    <option
value="9">Earth house</option>
    <option
value="10">Guest suite</option>
    <option
value="11">Guesthouse</option>
    <option
value="12">Hostel</option>
    <option
value="13">Hotel</option>
    <option
value="14">House</option>
    <option
value="15">Hut</option>
    <option
value="16">Loft</option>
    <option
value="17">Dorm</option>
    <option
value="18">Other</option>
    <option
value="19">Ryokan</option>
    <option
value="20">Serviced apartment</option>
    <option
value="21">Bungalow</option>
    <option
value="22">Tiny house</option>

```

```

value="23">Town house</option>
value="24">Villa</option>
</select>
</div>

<div class="form-group">
<label>Room type</label>
<select name="room_type" id="room_type"
class="form-control" required="required">
<option value="0">Entire
room</option>
<option
value="1">Private room</option>
<option value="2">Hotel
room</option>
<option value="3">Shared
room</option>
</select>
</div>

<div class="form-group">
<label>Accommodates</label>
<input id="accommodates" class="form-
control" name="accommodates" placeholder="Enter How Many People
Will Stay" type="number ">
</div>

<div class="form-group">
<label>Bathrooms</label>
<input id="bathrooms" class="form-control"
name="bathrooms" placeholder="Enter The Number of Bathrooms"
required="required">
</div>

<div class="form-group">
<label>Bedrooms</label>
<input id="bedrooms" name="bedrooms"
class="form-control" placeholder="Enter The Number of Bedrooms"
required="required">
</div>

<div class="form-group">
<label>Beds</label>
<input id="beds" name="beds" class="form-
control" placeholder="Enter The Number of Beds"
required="required">
</div>

<div class="form-group">
<label>Bed type</label>
<select name="bed_type" id="bed_type"
class="form-control" required="required">
<option value="0">Air
bed</option>

```

```

value="1">Futon</option>
value="2">Pull-out sofa</option>
bed</option>
</select>
</div>

<div class="form-group">
<label>Cleaning fee</label>
<input id="cleaning_fee"
name="cleaning_fee" class="form-control" placeholder="Enter
Additional Cleaning Fee in Yen" required="required">
</div>

<div class="form-group">
<label>Guests Included</label>
<input id="guests_included"
name="guests_included" class="form-control" placeholder="Enter How
Many Guest" required="required">
</div>

<div class="form-group">
<label>Extra People</label>
<input id="extra_people"
name="extra_people" class="form-control" placeholder="Enter
Additional Fees for Extra People" required="required">
</div>

<div class="form-group">
<label>Minimum Nights</label>
<input id="minimum_nights"
name="minimum_nights" class="form-control" placeholder="Enter
Minimum Nights Will Stay" required="required">
</div>

<button id="sub" type="submit" class="btn btn-
block btn-warning" >Check Price</button>
</form>
</div>

<h3 class="result">{{ prediction_text }}</h3>

</div>
</div>

</body>
</html>

```

Lampiran 4. *Output Pohon Random Forest*

