SKRIPSI

PREDIKSI HARGA TIKET PESAWAT MASKAPAI PENERBANGAN DI INDONESIA MENGGUNAKAN *TREE-BASED MODELS*

Ditulis untuk memenuhi sebagian persyaratan akademik
guna memperoleh gelar Sarjana Matematika

Oleh:

NAMA : KANAYA TABHITA DJIE

NPM : 01112180013

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PELITA HARAPAN
JAKARTA

2023

SKRIPSI

PREDIKSI HARGA TIKET PESAWAT MASKAPAI PENERBANGAN DI INDONESIA MENGGUNAKAN *TREE-BASED MODELS*

Ditulis untuk memenuhi sebagian persyaratan akademik
guna memperoleh gelar Sarjana Matematika

Oleh:

NAMA : KANAYA TABHITA DJIE

NPM : 01112180013

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PELITA HARAPAN
JAKARTA

2023

No. Dok. : TEM03/PRO12/STA04/SPMI-UPH	Rev : 02/17022020

UNIVERSITAS PELITA HARAPAN

Pernyataan dan Persetujuan Unggah Tugas Akhir

Saya/kami yang bertanda tangan di bawah ini:

Nama - NPM	:	1. Kanaya Tabhita Djie - 01112180013 2. 3.
Fakultas	:	Sains dan Teknologi
Program Studi	:	Matematika
Lokasi Kampus	:	Tangerang
Jenis Tugas Akhir	:	Skripsi
Judul	:	

PREDIKSI HARGA TIKET PESAWAT MASKAPAI PENERBANGAN DI INDONESIA MENGGUNAKAN *TREE-BASED MODELS*

Menyatakan bahwa:

1. Tugas akhir tersebut adalah benar karya saya/kami dengan arahan dari dosen pembimbing dan bukan merupakan duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai untuk mendapatkan gelar akademik di perguruan tinggi manapun;
2. Tugas akhir tersebut bukan merupakan plagiat dari hasil karya pihak lain, dan apabila saya/kami mengutip dari karya orang lain maka akan dicantumkan sebagai referensi sesuai dengan ketentuan yang berlaku.
3. Saya/kami memberikan Hak Noneksklusif Tanpa Royalti kepada Universitas Pelita Harapan atas Tugas Akhir tersebut untuk diunggah ke dalam Repositori UPH.

Apabila di kemudian hari ditemukan pelanggaran Hak Cipta dan Kekayaan Intelektual atau Peraturan Perundangan-undangan Republik Indonesia lainnya dan integritas akademik dalam karya saya/kami tersebut, maka saya/kami bersedia menanggung secara pribadi segala bentuk tuntutan hukum dan sanksi akademis yang timbul serta membebaskan Universitas Pelita Harapan dari segala tuntutan hukum yang berlaku.

Dibuat di : Tangerang
Pada Tanggal : 23-Jan-2023
Yang menyatakan,

Tanda Tangan	METERAI ELEKTRONIK 10000 SEPULUH RIBU RUPIAH 6000	*(tanpa meterai)*	*(tanpa meterai)*
Nama	(Kanaya Tabhita Djie - 01112180013)	( )	( )

UNIVERSITAS PELITA HARAPAN
FAKULTAS SAINS DAN TEKNOLOGI

PERSETUJUAN DOSEN PEMBIMBING SKRIPSI

PREDIKSI HARGA TIKET PESAWAT

MASKAPAI PENERBANGAN DI INDONESIA

MENGGUNAKAN *TREE-BASED MODELS*

Oleh:

Nama	: **Kanaya Tabhita Djie**
NPM	: **01112180013**
Program Studi	: **Matematika**
Peminatan/Konsentrasi	: **Matematika Bisnis**

telah dipertahankan dalam Sidang Skripsi dan disetujui oleh dosen pembimbing skripsi guna memperoleh gelar Sarjana Matematika pada Program Studi Matematika, Fakultas Sains dan Teknologi Universitas Pelita Harapan, Jakarta.

Tangerang, 23 Januari 2023

Menyetujui,

Pembimbing Utama	**Pembimbing Pendamping**
(Lina Cahyadi, M.Si.)	(Josephine, M.Math.)

Mengetahui,

Ketua Program Studi Matematika	**Dekan**
(Kie Van Ivanky Saputra, Ph.D.)	(Eric Jobiliong, Ph.D.)

UNIVERSITAS PELITA HARAPAN
FAKULTAS SAINS DAN TEKNOLOGI

PERSETUJUAN TIM PENGUJI SKRIPSI

Pada hari Jumat, 6 Januari 2023 telah diselenggarakan Sidang Skripsi untuk memenuhi sebagian persyaratan akademik guna memperoleh gelar Sarjana Matematika pada Program Studi Matematika, Fakultas Sains dan Teknologi Universitas Pelita Harapan, atas nama:

Nama	**: Kanaya Tabhita Djie**
NPM	**: 01112180013**
Program Studi	**: Matematika**
Fakultas	**: Sains dan Teknologi**

dengan Skripsi yang berjudul "PREDIKSI HARGA TIKET PESAWAT MASKAPAI PENERBANGAN DI INDONESIA MENGGUNAKAN *TREE-BASED MODELS*" dan telah berhasil dipertahankan dan disetujui oleh tim penguji yang terdiri dari:

Nama Penguji	**Jabatan dalam Tim Penguji**	**Tanda Tangan**
1. Lina Cahyadi, M.Si.,	sebagai Ketua	
2. Josephine, M.Math.,	sebagai Anggota	
3. Dr. Ir. Samuel Lukas, M. Tech.,	sebagai Anggota	

ABSTRAK

Kanaya Tabhita Djie (01112180013)

PREDIKSI HARGA TIKET PESAWAT MASKAPAI PENERBANGAN DI INDONESIA MENGGUNAKAN *TREE-BASED MODELS*

Skripsi, Fakultas Sains dan Teknologi (2023)

(xvi + 92 halaman, 47 gambar, 25 tabel, 2 lampiran)

Setiap harinya, data yang dihasilkan pada industri penerbangan Indonesia sangat besar. Data yang tersedia seperti jam keberangkatan, jam kedatangan, jumlah transit, hari penerbangan, dan sebagainya dapat digunakan untuk memprediksi harga tiket pesawat. Penelitian ini menggunakan tiga metode *machine learning* yaitu *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* untuk mendapatkan hasil prediksi harga tiket pesawat pada rute Jakarta-Surabaya dan Jakarta-Manado. Untuk setiap model prediksi akan dilakukan *parameter tuning* untuk mendapatkan model yang paling optimal. Setelah didapatkan tiga model yang optimal, dilakukan evaluasi model untuk menentukan model terbaik untuk setiap rute. Evaluasi model dilakukan dengan membandingkan nilai *Mean Absolute Error*, *Mean Absolute Percentage Error*, *Root Mean Squared Error*, dan Koefisien Determinasi dari tiap model. Hasil penelitian menunjukkan bahwa model yang memiliki akurasi paling baik untuk kedua rute tersebut adalah model *Random Forest*.

Kata Kunci : *regression tree*, *bagging regression tree*, *random forest*, *machine learning*

Referensi : 26 (1967-2021)

ABSTRACT

Kanaya Tabhita Djie (01112180013)

FLIGHT FARE IN INDONESIA PREDICTION USING TREE-BASED MODELS

Thesis, Faculty of Science and Technology (2023)

(xvi + 92 pages, 47 figures, 25 tables, 2 appendices)

Every day, the data generated from the aviation industry in Indonesia is enormous. Available data such as departure time, arrival time, number of transits, scheduled flight day, and so on may be used to predict flight fares. This study uses three machine learning techniques, namely Regression Tree, Bagging Regression Tree, and Random Forest to get the flight fares prediction result for Jakarta-Surabaya and Jakarta-Manado routes. For each prediction model, parameter tuning will be carried out to obtain the most optimal models. After obtaining three optimal models, model evaluation is performed to determine the best model for each route. Model evaluation is done by comparing each value of the Mean Absolute Error, Mean Absolute Percentage Error, Root Mean Squared Error and Coefficient of Determination of each model. The results reveal Random Forest as the model with the best accuracy for both routes.

Keywords : regression tree, bagging regression tree, random forest, machine learning

Reference : 26 (1967-2021)

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan laporan skripsi dengan judul "PREDIKSI HARGA TIKET PESAWAT MASKAPAI PENERBANGAN DI INDONESIA MENGGUNAKAN *TREE-BASED MODELS*" dengan baik dan tepat pada waktunya.

Laporan skripsi ini disusun berdasarkan penelitian yang telah dilakukan dari bulan tahun mulai hingga bulan tahun akhir. Skripsi merupakan persyaratan terakhir bagi mahasiswa yang wajib ditempuh sesuai dengan kurikulum Program Studi Matematika Fakultas Sains dan Teknologi, Universitas Pelita Harapan. Skripsi ini juga bermanfaat bagi penulis untuk menerapkan pengetahuan yang telah didapat dan memperoleh pengalaman baru yang tidak dapat diperoleh dari perkuliahan.

Dalam penyusunan laporan skripsi ini, penulis mendapat dukungan dari banyak pihak. Oleh karena itu, saya mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Eric Jobiliong, Ph.D., selaku Dekan Fakultas Sains dan Teknologi;
2. Ibu Dr. Nuri Arum Anugrahati, selaku Wakil Dekan Fakultas Sains dan Teknologi;
3. Bapak Laurence, M.T., selaku Direktur Administrasi dan Kemahasiswaan Fakultas Sains dan Teknologi;
4. Bapak Kie Van Ivanky Saputra, Ph.D., selaku Ketua Program Studi Matematika;
5. Ibu Lina Cahyadi, M.Si., selaku pembimbing skripsi yang telah memberikan masukan, arahan, dan dukungan dalam pengerjaan laporan skripsi;
6. Ibu Josephine, M.Math., selaku co-pembimbing skripsi yang telah memberikan masukan, arahan, dan dukungan dalam pengerjaan laporan skripsi;
7. Bapak Ferry Vincenttius Ferdinand, S.Si., S.Inf., M.Pd., M.M., selaku dosen

pembimbing akademik yang telah membantu penulis selama perkuliahan.

8. seluruh dosen yang telah mendidik dan membagikan ilmunya kepada penulis selama masa perkuliahan;
9. keluarga penulis yang telah memberikan banyak dukungan dan semangat dari awal hingga akhir perkuliahan;
10. Sindy Sintya, Vanessa Laurencia, Veronica Cynthia, dan Alvin Lorenz Golton yang telah menemani dan mendukung seluruh proses perkuliahan dan penulisan skripsi;
11. Vivien Carin selaku sahabat baik dari penulis yang selalu ada untuk memberi dukungan dan semangat di masa sulit;
12. teman-teman Matematika 2018 atas kebersamaannya selama 4 tahun;
13. HMM-UPH 2019/2020, yaitu Alvin, Dea, Bernath, Cheryl, Teo, Lou, Fannie, Felix, Gabrielle, Gaby, Joey, Licya, Marcelina, Michelle, Monica, Revata, Sindy, Stephanie, Terry, Vanessa, Veren, Vero, dan Yudi;
14. keluarga Mathies yang telah menjadi keluarga baru selama menjalani dunia perkuliahan; dan
15. pihak lain yang telah membantu penulis dalam proses perkuliahan dan penulisan skripsi secara langsung maupun tidak langsung yang tidak dapat disebutkan satu persatu.

Akhir kata, penulis menyadari bahwa laporan skripsi ini masih sangat jauh dari kata sempurna. Oleh karena itu, penulis sangat terbuka akan kritik dan saran dari pembaca yang dapat membantu membuat laporan skripsi ini menjadi lebih baik lagi. Semoga laporan ini dapat bermanfaat bagi para pembacanya.

Tangerang, 23 Januari 2023

(Kanaya Tabhita Djie)

DAFTAR ISI

halaman

HALAMAN JUDUL
PERNYATAAN DAN PERSETUJUAN UNGGAH TUGAS AKHIR
PERSETUJUAN DOSEN PEMBIMBING SKRIPSI
PERSETUJUAN TIM PENGUJI SKRIPSI
ABSTRAK v
ABSTRACT vi
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xiii
DAFTAR TABEL xv
DAFTAR LAMPIRAN xvi

BAB I PENDAHULUAN
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan Penelitian 3
1.4 Batasan Masalah 3
1.5 Manfaat Penelitian 4
1.5.1 Manfaat Teoritis 4
1.5.2 Manfaat Praktis 4
1.6 Sistematika Penulisan 4

BAB II LANDASAN TEORI
2.1 *Machine Learning* 6
2.1.1 *Big Data* 6
2.1.2 Pengertian *Machine Learning* dan Masalah yang Dapat Diselesaikan 7
2.1.3 *Supervised Learning* dan *Unsupervised Learning* 8
2.1.4 Proses *Machine Learning* 9
2.2 *Tree-Based Models* 10
2.2.1 Pengertian *Tree-Based Models* 10
2.2.2 Terminologi 11
2.2.3 Struktur *Tree-Based Models* 12
2.3 *Regression Tree* 13
2.3.1 Proses Membangun Model *Regression Tree* 13
2.4 *The Bias-Variance Trade Off* 15
2.5 *Overfitting* dan *Underfitting* 17
2.5.1 *Overfitting* 17
2.5.2 *Underfitting* 18
2.5.3 *Fit* yang Baik pada *Machine Learning* 18

2.6 Meningkatkan Akurasi *Regression Tree* 19
2.6.1 *Pruning* 19
2.6.2 *Bagging* 21
2.6.3 *Random Forest* 22
2.6.4 *Hyperparameter Optimization* 23
2.7 *k-Fold Cross-Validation* 23
2.8 Evaluasi Model 24
2.8.1 *Root Mean Squared Error* 24
2.8.2 *Mean Absolute Error* 25
2.8.3 *Mean Absolute Percentage Error* 25
2.8.4 Koefisien Determinasi 26
2.9 Tinjauan Pustaka 26

BAB III METODOLOGI
3.1 Pengumpulan Data 29
3.2 Analisis Data 32
3.3 Pembagian Data 32
3.4 Model Prediksi dengan *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* 33
3.5 Memprediksi Data 35
3.6 Evaluasi Model 35
3.7 *Web App* dengan *R Shiny* 36

BAB IV ANALISIS DAN PEMBAHASAN
4.1 Data 37
4.2 Analisis Data 39
4.2.1 Rute Jakarta-Surabaya (CGK-SUB) 40
4.2.1.1 Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Surabaya 40
4.2.1.2 Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Surabaya . . . 40
4.2.1.3 Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Surabaya 41
4.2.1.4 Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Surabaya 42
4.2.1.5 Harga Tiket Pesawat Berdasarkan Hari Penerbangan pada Rute Jakarta-Surabaya 43
4.2.1.6 Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Surabaya 44
4.2.1.7 Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Surabaya 45
4.2.1.8 Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Surabaya 46

4.2.1.9 Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya 47
4.2.1.10 Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya 48
4.2.2 Rute Jakarta-Manado (CGK-MDC) 49
4.2.2.1 Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Manado 49
4.2.2.2 Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Manado 50
4.2.2.3 Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Manado 51
4.2.2.4 Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Manado 52
4.2.2.5 Harga Tiket Pesawat Berdasarkan Hari Penerbangan pada Rute Jakarta-Manado 53
4.2.2.6 Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Manado 53
4.2.2.7 Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Manado 54
4.2.2.8 Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Manado 55
4.2.2.9 Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Manado 56
4.2.2.10 Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Manado 56
4.3 Pembagian Data 57
4.4 Model Prediksi untuk Rute Jakarta-Surabaya 57
4.4.1 *Regression Tree* 57
4.4.1.1 Model *Regression Tree* Awal 57
4.4.1.2 Model *Tuned Regression Tree* 60
4.4.1.3 Performa Model 62
4.4.2 *Bagging Regression Tree* 62
4.4.2.1 Model *Bagging Regression Tree* 62
4.4.2.2 Performa Model 65
4.4.3 *Random Forest* 65
4.4.3.1 Model *Random Forest* Awal 65
4.4.3.2 Model *Tuned Random Forest* 66
4.4.3.3 Performa Model 70
4.4.4 Evaluasi Performa Seluruh Model 70

4.5 Model Prediksi untuk Rute Jakarta-Manado 72
4.5.1 *Regression Tree* 72
4.5.1.1 Model *Regression Tree* Awal 72
4.5.1.2 Model *Tuned Regression Tree* 75
4.5.1.3 Performa Model 77
4.5.2 *Bagging Regression Tree* 78
4.5.2.1 Model *Bagging Regression Tree* 78
4.5.2.2 Performa Model 80
4.5.3 *Random Forest* 81
4.5.3.1 Model *Random Forest* Awal 81
4.5.3.2 Model *Tuned Random Forest* 82
4.5.3.3 Performa Model 85
4.5.4 Evaluasi Performa Seluruh Model 85
4.6 *Web App* Prediksi Harga Tiket Pesawat Menggunakan *R Shiny* 87

BAB V PENUTUP
5.1 Kesimpulan 89
5.2 Saran 89

DAFTAR PUSTAKA 90

LAMPIRAN

DAFTAR GAMBAR

halaman

Gambar 2.1 Komponen Utama *Big Data* 6
Gambar 2.2 *Supervised Learning* 8
Gambar 2.3 *Unsupervised Learning* 9
Gambar 2.4 Proses *Machine Learning* 9
Gambar 2.5 Proses Segmentasi Data dan Bentuk Diagram Pohon pada *Tree-Based Models* 11
Gambar 2.6 Struktur Dasar *Tree-Based Models* 12
Gambar 2.7 *Overfitting* pada *Decision Trees* 19

Gambar 3.1 *Flowchart* Langkah-Langkah Pengerjaan 28
Gambar 3.2 Contoh *Splitting* Pertama 34

Gambar 4.1 Contoh Data Penerbangan Rute Jakarta-Surabaya 37
Gambar 4.2 Contoh Data Penerbangan Rute Jakarta-Manado 38
Gambar 4.3 Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Surabaya 40
Gambar 4.4 Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Surabaya 41
Gambar 4.5 Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Surabaya 42
Gambar 4.6 Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Surabaya 43
Gambar 4.7 Harga Tiket Pesawat Berdasarkan Hari Penerbangan pada Rute Jakarta-Surabaya 44
Gambar 4.8 Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Surabaya 45
Gambar 4.9 Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Surabaya 46
Gambar 4.10 Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Surabaya 47
Gambar 4.11 Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya 48
Gambar 4.12 Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya . . . 49
Gambar 4.13 Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Manado 50
Gambar 4.14 Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Manado 51
Gambar 4.15 Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Manado 52

Gambar 4.16 Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Manado 52
Gambar 4.17 Harga Tiket Pesawat Berdasarkan Hari Keberangkatan pada Rute Jakarta-Manado 53
Gambar 4.18 Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Manado 54
Gambar 4.19 Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Manado 55
Gambar 4.20 Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Manado 55
Gambar 4.21 Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Manado 56
Gambar 4.22 Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Manado 57
Gambar 4.23 *Plot* Model *Regression Tree* Awal untuk Rute Jakarta-Surabaya 59
Gambar 4.24 *Plot* CP Model *Regression Tree* Awal untuk Rute Jakarta-Surabaya 60
Gambar 4.25 *Plot* Jumlah Model Pohon terhadap Nilai Eror pada Model *Bagging Regression Tree* untuk Rute Jakarta-Surabaya . . . 63
Gambar 4.26 Urutan *Variable Importance* pada Model *Bagging Regression Tree* untuk Rute Jakarta-Surabaya 64
Gambar 4.27 *Plot* Hasil *Parameter Tuning* terhadap Parameter *mtry* pada Model *Random Forest* untuk Rute Jakarta-Surabaya 68
Gambar 4.28 Visualisasi Performa Model untuk Rute Jakarta-Surabaya . 71
Gambar 4.29 Nilai Varians Model Prediksi untuk Rute Jakarta-Surabaya . 72
Gambar 4.30 *Plot* Model *Regression Tree* Awal untuk Rute Jakarta-Manado 74
Gambar 4.31 *Plot* CP Model *Regression Tree* Awal untuk Rute Jakarta-Manado 75
Gambar 4.32 *Plot* Jumlah Model Pohon terhadap Nilai Eror pada Model *Bagging Regression Tree* untuk Rute Jakarta-Manado . . . 78
Gambar 4.33 Urutan *Variable Importance* pada Model *Bagging Regression Tree* untuk Rute Jakarta-Manado 80
Gambar 4.34 *Plot* Hasil *Parameter Tuning* terhadap Parameter *mtry* pada Model *Random Forest* untuk Rute Jakarta-Manado 83
Gambar 4.35 Visualisasi Performa Model untuk Rute Jakarta-Manado . . 86
Gambar 4.36 Nilai Varians Model Prediksi untuk Rute Jakarta-Manado . 87
Gambar 4.37 *Navigation Bar Web App* Prediksi Harga Tiket Pesawat . . 87
Gambar 4.38 *Web App* Prediksi Harga Tiket Pesawat 88

DAFTAR TABEL

halaman

Tabel 2.1 Interpretasi Nilai MAPE 26

Tabel 3.1 Data Mentah 29
Tabel 3.2 Data Bersih 30
Tabel 3.3 Variabel pada Data Penerbangan 32
Tabel 3.4 Contoh Data Penerbangan 33
Tabel 3.5 Contoh Pembagian Data Bagian Pertama 33
Tabel 3.6 Contoh Pembagian Data Bagian Kedua 34
Tabel 3.7 Contoh Data Penerbangan dan Hasil Prediksinya 35

Tabel 4.1 Kelas dan Elemen untuk Setiap Variabel pada Data Penerbangan 39
Tabel 4.2 Tingkat Kesalahan Prediksi pada Setiap *Terminal Nodes* Model *Regression Tree* untuk Rute Jakarta-Surabaya 62
Tabel 4.3 Performa Model *Regression Tree* untuk Rute Jakarta-Surabaya 62
Tabel 4.4 Performa Model *Bagging Regression Tree* untuk Rute Jakarta-Surabaya 65
Tabel 4.5 *Parameter Tuning* terhadap Parameter *ntree* pada Model *Random Forest* untuk Rute Jakarta-Surabaya 68
Tabel 4.6 *Parameter Tuning* terhadap Parameter *maxnodes* pada Model *Random Forest* untuk Rute Jakarta-Surabaya 69
Tabel 4.7 Performa Model *Random Forest* untuk Rute Jakarta-Surabaya 70
Tabel 4.8 Performa Model untuk Rute Jakarta-Surabaya 70
Tabel 4.9 Nilai Bias dan Varians Model Prediksi untuk Rute Jakarta-Surabaya 71
Tabel 4.10 Tingkat Kesalahan Prediksi pada Setiap *Terminal Nodes* Model *Regression Tree* untuk Rute Jakarta-Manado 77
Tabel 4.11 Performa Model *Regression Tree* untuk Rute Jakarta-Manado 77
Tabel 4.12 Performa Model *Bagging Regression Tree* untuk Rute Jakarta-Manado 80
Tabel 4.13 *Parameter Tuning* terhadap Parameter *ntree* pada Model *Random Forest* untuk Rute Jakarta-Manado 83
Tabel 4.14 *Parameter Tuning* terhadap Parameter *maxnodes* pada Model *Random Forest* untuk Rute Jakarta-Manado 84
Tabel 4.15 Performa Model *Random Forest* untuk Rute Jakarta-Manado . 85
Tabel 4.16 Performa Model untuk Rute Jakarta-Manado 85
Tabel 4.17 Nilai Bias dan Varians Model Prediksi untuk Rute Jakarta-Manado 86

DAFTAR LAMPIRAN

halaman

Lampiran A Data

A.1 Data Penerbangan Rute Jakarta-Surabaya A-1

A.2 Data Penerbangan Rute Jakarta-Manado A-4

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan sehari-hari, manusia tidak akan terlepas dari data. Bahkan sejak lahir, manusia sudah menghasilkan data seperti nama, tanggal lahir, jam lahir, berat badan, dan sebagainya. Sagiroglu & Sinanc [1] menyatakan bahwa jumlah data yang telah dihasilkan manusia pada tahun 2003 adalah sebesar 5 *exabit* (10^{18} bit) sedangkan pada tahun 2013, jumlah data ini dapat dihasilkan hanya dalam waktu dua hari. Fenomena data yang melimpah ini dikenal sebagai *big data*, yaitu sekumpulan data terstruktur maupun tidak terstruktur yang volumenya sangat besar. Pengolahan *big data* akan sangat sulit jika menggunakan metode yang tradisional, sehingga dapat digunakan aplikasi dari *artificial intelligence* yaitu *machine learning* (ML).

Wang & Alexander [2] menjelaskan bahwa ML dapat dimanfaatkan untuk memperoleh informasi dan wawasan yang akan digunakan untuk membuat keputusan tertentu. Algoritma ML dapat diklasifikasikan menjadi dua kategori, yaitu *supervised learning* dan *unsupervised learning*. Beberapa contoh metode yang menggunakan algoritma *supervised learning* adalah *Naïve Bayes* dan *Support Vector Machine* (SVM). Berbeda dengan algoritma *supervised learning* yang menggunakan data yang sudah memiliki label, *unsupervised learning* menggunakan data yang tidak memiliki label dan melakukan klasifikasi dengan membandingkan fitur dari data tersebut. Metode *Clustering* adalah salah satu contoh metode algoritma *unsupervised learning*.

Menurut Clark & Pregibon [3], *tree-based models* adalah metode *supervised learning* yang digunakan untuk menemukan struktur di data. Metode ini menggunakan beberapa pernyataan bersyarat untuk mempartisi data menjadi beberapa himpunan bagian. Beberapa contoh dari *tree-based models* adalah *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*. Prasad *et al.* [4] menjelaskan perbedaan antara *Regression Tree* dengan teknik regresi klasik. Berbeda dengan teknik regresi klasik di mana hubungan antara variabel prediktor dan variabel responsnya sudah ditentukan seperti linear atau kuadratik, *Regression Tree* tidak mengasumsikan hubungan apa pun. *Output* dari metode tersebut berupa diagram pohon dengan cabang-cabang yang ditentukan dari aturan *splitting* dan sejumlah *terminal nodes* yang merupakan rata-rata dari variabel respons. ML

dapat dimanfaatkan di berbagai industri dan salah satunya adalah di industri penerbangan. Industri penerbangan di Indonesia menghasilkan data dengan volume yang sangat besar setiap harinya.

Fatimah [5] menjelaskan bahwa Indonesia adalah negara kepulauan yang mencakup lima pulau besar, ratusan pulau sedang, dan ribuan pulau kecil. Luas wilayah Indonesia yang sangat besar juga mengakibatkan diperlukannya sarana transportasi yang dapat memadai kebutuhan mobilisasi banyak orang dari satu tempat menuju ke tempat yang lainnya, sehingga tidak jarang orang yang memanfaatkan sarana transportasi udara untuk melakukan kegiatan bepergian. Oleh karena itu, perusahaan maskapai penerbangan tentunya menginginkan pendapatan yang maksimum sebagai jasa penyedia transportasi udara tersebut. Untuk mencapai target pendapatan tersebut, perusahaan menggunakan *revenue management* untuk menentukan harga tiket pesawat yang paling optimum. Menurut Li & Peng [6], dalam menentukan harga tiket penerbangan, perusahaan mempertimbangkan berbagai banyak hal seperti tingkat permintaan dan *inventory level*. Data ini tidak terbuka bagi umum dan sulit bagi calon penumpang untuk memprediksi harga tiket pesawat karena ketidaktersediaannya data tersebut.

Untuk mengatasi masalah ini, dapat dimanfaatkan data penerbangan yang terbuka untuk umum sebagai faktor untuk memprediksi harga pesawat. Data yang dapat digunakan untuk membangun model prediksi yaitu jam kedatangan, jam keberangkatan, jumlah transit, hari penerbangan, dan sebagainya. Algoritma *tree-based models* yaitu *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* adalah metode yang dipilih untuk memprediksi harga tiket pesawat. Ketiga metode tersebut juga akan dibandingkan akurasinya untuk mendapatkan model yang terbaik. Rute penerbangan domestik di Indonesia yaitu Jakarta-Surabaya dan Jakarta-Manado adalah salah satu rute dengan jumlah penerbangan yang cukup banyak yaitu 20-30 penerbangan setiap harinya. Kedua rute ini akan digunakan untuk membuat model prediksi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah pada penelitian ini adalah sebagai berikut.

1. Bagaimana cara memprediksi harga tiket pesawat maskapai penerbangan di Indonesia pada rute Jakarta-Surabaya dan Jakarta-Manado menggunakan

metode *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*?

2. Bagaimana perbandingan akurasi metode *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* dalam memprediksi harga tiket pesawat maskapai penerbangan di Indonesia pada rute Jakarta-Surabaya dan Jakarta-Manado?

1.3 Tujuan Penelitian

Berawal dari rumusan masalah yang akan dipecahkan, penelitian ini memiliki dua tujuan utama yaitu sebagai berikut.

1. Memperoleh prediksi harga tiket pesawat maskapai penerbangan di Indonesia pada rute Jakarta-Surabaya dan Jakarta-Manado dengan metode *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*.
2. Membandingkan keakuratan metode *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* dalam memprediksi harga tiket pesawat maskapai penerbangan di Indonesia pada rute Jakarta-Surabaya dan Jakarta-Manado.

1.4 Batasan Masalah

Berikut adalah batasan masalah beserta asumsi yang dipakai untuk mencapai tujuan penelitian.

1. Data yang digunakan adalah seluruh data penerbangan dari 9 Juni 2022 hingga 31 Agustus 2022.
2. Data penerbangan dikumpulkan setiap hari dari situs web *tiket.com* menggunakan *browser extension* yaitu *Web Scraper*.
3. Rute penerbangan yang digunakan dalam penelitian adalah rute penerbangan domestik yaitu Jakarta-Surabaya (CGK-SUB) dan Jakarta-Manado (CGK-MDC).
4. Maskapai penerbangan yang digunakan dalam penelitian adalah maskapai penerbangan di Indonesia yaitu Batik Air, Citilink, Garuda Indonesia, Lion Air, Nam Air, dan Super Air Jet.
5. Kelas kabin yang digunakan dalam penelitian ini adalah ekonomi.

6. Zona waktu yang digunakan dalam penelitian adalah Waktu Indonesia Barat (WIB).
7. Metode *tree-based models* yang digunakan adalah *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*.
8. Bahasa pemrograman yang digunakan adalah *R*.
9. *Package R* yang digunakan untuk metode *Regression Tree* adalah *rpart* [7].
10. *Package R* yang digunakan untuk metode *Bagging Regression Tree* adalah *caret* [8].
11. *Package R* yang digunakan untuk metode *Random Forest* adalah *caret* [8].

1.5 Manfaat Penelitian

Pada penelitian ini, terdapat dua jenis manfaat yaitu manfaat teoritis dan manfaat praktis. Berikut dijelaskan beberapa manfaat yang bisa diambil dari penelitian ini.

1.5.1 Manfaat Teoritis

Pada penelitian ini, manfaat teoritisnya adalah untuk menemukan metode *tree-based models* yang paling optimal untuk memprediksi harga tiket pesawat maskapai penerbangan di Indonesia pada rute Jakarta-Surabaya dan Jakarta-Manado.

1.5.2 Manfaat Praktis

Pada penelitian ini, manfaat praktis yang dapat diambil adalah mengurangi ketidakpastian harga tiket pesawat dan dapat digunakan calon penumpang pesawat untuk menentukan jadwal penerbangan yang diinginkan dengan harga paling optimal.

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan skripsi yang diterapkan untuk penulisan dari penelitian ini.

1. BAB I - Pendahuluan

   Latar belakang, rumusan masalah, tujuan penelitian, batasan dan asumsi penelitian, manfaat penelitian, dan sistematika penulisan penelitian ini akan dijelaskan pada Bab I.

2. BAB II - Landasan Teori

   Pada bab ini dijelaskan beberapa teori yang akan digunakan untuk mendukung penulisan penelitian. Teori yang dibahas antara lain *machine learning*, *tree-based models*, *Regression Tree*, *bias and variance trade-Off*, *overfitting and underfitting*, cara meningkatkan akurasi *Regression Tree*, *k-fold cross-validation* dan evaluasi model. Selain itu, tinjauan pustaka yang dimanfaatkan sebagai referensi bagi penelitian ini juga dimuat dalam bab ini.

3. BAB III - Metodologi

   Pada Bab III dijelaskan proses atau langkah pengerjaan yang akan dilaksanakan secara rinci.

4. BAB IV - Analisis dan Pembahasan

   Hasil yang diperoleh dari pengolahan data menggunakan metode *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*, beserta perbandingan akurasi setiap model dijelaskan pada bab IV. Hasil analisis tersebut berupa model yang terbaik untuk memprediksi harga tiket pesawat.

5. BAB V - Kesimpulan dan Saran

   Pada bab ini dijelaskan penarikan kesimpulan yang didapatkan dari hasil penelitian, beserta saran-saran untuk pengembangan penelitian yang serupa kedepannya.

BAB II

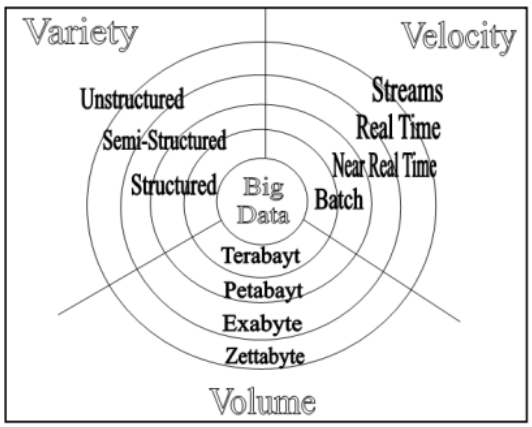
LANDASAN TEORI

Pada bab ini akan dijelaskan lebih lanjut mengenai teori-teori dan tinjauan pustaka yang akan digunakan dalam penelitian ini.

2.1 *Machine Learning*

2.1.1 *Big Data*

Sagiroglu dan Sinanc [1] menyatakan bahwa *big data* adalah istilah untuk menggambarkan sekumpulan data yang memiliki struktur yang lebih besar, beragam, dan kompleks disertai dengan kerumitan dalam penyimpanan, analisis, dan visualisasi untuk proses atau hasil yang lebih lanjut. Dalam proses penelitian, volume data yang besar ini dapat menunjukkan berbagai informasi seperti pola atau korelasi. Informasi ini sangat bermanfaat karena dapat digunakan untuk memperoleh wawasan yang lebih dalam mengenai suatu hal. Karakteristik *big data* dapat digambarkan dengan tiga komponen utama yaitu *variety* (variasi), *velocity* (kecepatan), dan volume seperti yang ditunjukkan pada Gambar 2.1.



Gambar 2.1: Komponen Utama *Big Data*
Sumber: *Big Data: A Review* [1]

Keragaman menyebabkan ukuran data menjadi sangat besar. Pada umumnya, *big data* terdiri dari tiga jenis yaitu *structured data*, *semi-structured data*, dan *unstructured data*. *Structured data* adalah data yang telah terorganisir sehingga elemen-elemennya sudah disimpan sesuai dengan format yang telah

ditentukan, misalnya seperti *database*. *Unstructured data* bersifat acak dan sulit untuk dianalisis. *Semi-structured data* memiliki beberapa struktur namun tidak sekaku *structured data*. Volume data sekarang sudah melampaui *terabytes* dan *petabytes*. Jumlah ini sangat besar dan akan terus meningkat. Data yang dihasilkan dapat dihasilkan secara terus menerus menyerupai sebuah aliran (*streams)*.

2.1.2 Pengertian *Machine Learning* dan Masalah yang Dapat Diselesaikan

Menurut Alzubi *et al.* [9], *machine learning* adalah salah satu kategori dari *artificial intelligence* yang memberikan kemampuan pada komputer untuk berpikir dan belajar dengan sendirinya, sehingga perubahan tindakan yang lebih baik dapat dilakukan untuk meningkatkan akurasi oleh komputer. Istilah *machine learning* diciptakan oleh Arthur Samuel [10] pada tahun 1959, yang mendefinisikan ML sebagai sebuah bidang studi yang memberikan kemampuan belajar kepada komputer tanpa harus diprogram secara eksplisit. ML dapat melakukan analisis terhadap data yang jumlahnya besar dan kompleks secara efektif.

Alzubi *et al.* [9] juga menjelaskan bahwa terdapat suatu metode ML yang tepat untuk diterapkan ke masing-masing masalah yang ingin diselesaikan. Masalah-masalah tersebut dapat dibagi lima yaitu sebagai berikut.

1. *Classification Problem*
   Classification problem adalah masalah yang *output*-nya berupa salah satu dari beberapa kelas *output* yang jumlahnya tetap, seperti Ya/Tidak, *True*/*False*, atau 0/1. Berdasarkan jumlah kelas *output*, masalah ini dapat dibagi menjadi *binary* atau *multi-class classification problem*.

2. *Anomaly Detection Problem*
   Anomaly detection problem berkaitan dengan analisis terhadap pola tertentu dan mendeteksi perubahan atau anomali pada pola tersebut. Masalah ini biasanya berhubungan dengan pencarian *outlier*.

3. *Regression Problem*
   Regression problem berkaitan masalah yang *output*-nya kontinu atau numerik.

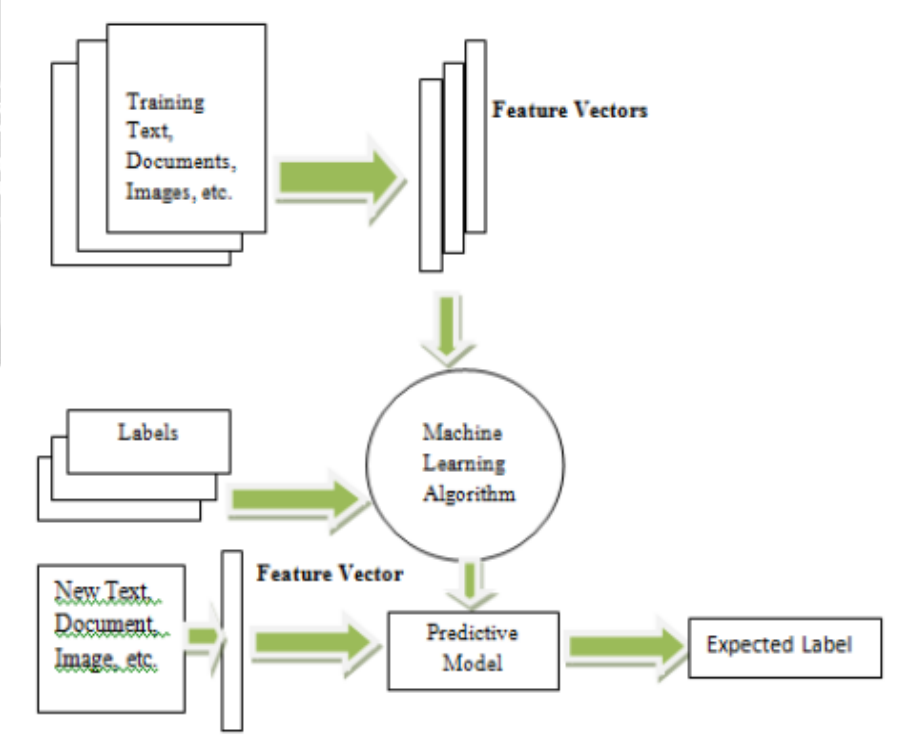
4. *Clustering problem*
   Clustering problem berkaitan dengan data yang tidak memiliki label dan perlu dikelompokkan berdasarkan kemiripan struktur data. Kelompok tersebut kemudian dapat diberi label.

5. *Reinforcement Problem*

   Reinforcement problem berkaitan dengan masalah yang dapat diselesaikan dengan sebuah keputusan yang dibuat dari pengalaman proses *learning* sebelumnya.

2.1.3 *Supervised Learning* dan *Unsupervised Learning*

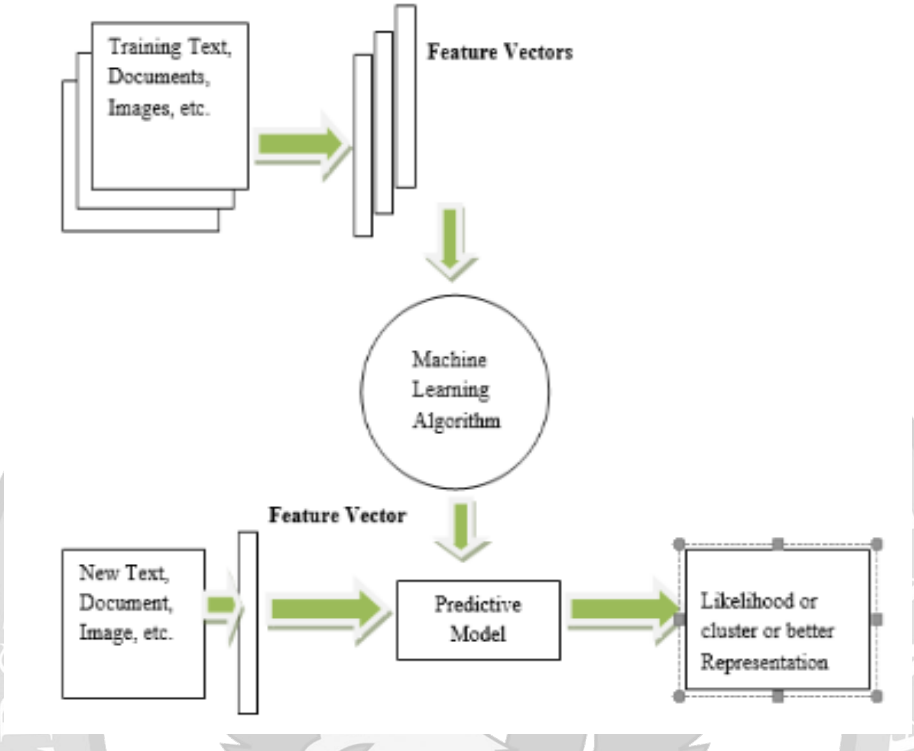
Algoritma ML dapat dibagi menjadi dua yaitu *supervised learning* dan *unsupervised learning*. Alzubi *et al.* [9] menjelaskan bahwa dalam *supervised learning*, data latihan (*training*) yang digunakan sudah tersedia dengan *output* yang benar. Algoritma akan belajar untuk memberikan respons yang lebih akurat berdasarkan *output* tersebut. *Supervised learning* dikenal juga dengan istilah "*learning from examples*". Konsep *supervised learning* dapat dijelaskan pada Gambar 2.2. Contoh algoritma *supervised learning* adalah *Decision Tree*, *Naiïve Bayes*, *Support Vector Machine* (SVM), dan Analisis Regresi.



Gambar 2.2: *Supervised Learning*
Sumber: *Machine Learning from Theory to Algorithms: An Overview* [9]

Algoritma *unsupervised learning* digunakan ketika kategori dari suatu data tidak diketahui, sehingga data latihan tidak memiliki label. Algoritma ini mencari

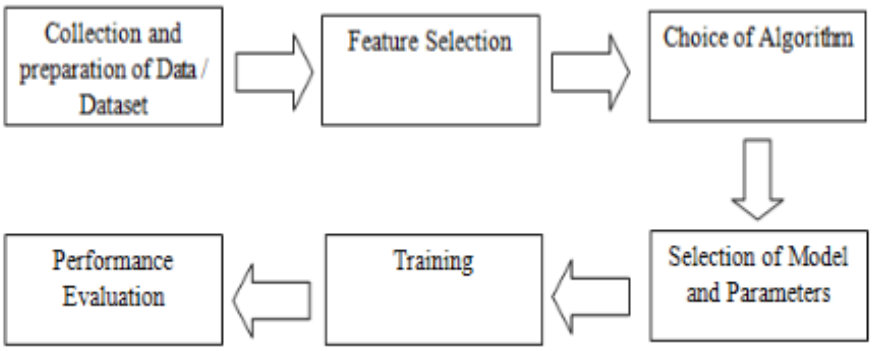
pola tersembunyi dari data dan membuat aturan pengelompokan dari data tersebut [9]. Konsep *unsupervised learning* dapat dijelaskan pada Gambar 2.3. Contoh algoritma *unsupervised learning* adalah *K-Means Clustering*.



Gambar 2.3: *Unsupervised Learning*
Sumber: *Machine Learning from Theory to Algorithms: An Overview* [9]

2.1.4 Proses *Machine Learning*

Proses pada ML dapat dlihat pada Gambar 2.4.



Gambar 2.4: Proses *Machine Learning*
Sumber: *Machine Learning from Theory to Algorithms: An Overview* [9]

Menurut Alzubi *et al.* [9], proses pada ML dapat dibagi menjadi enam yaitu sebagai berikut.

1. Pengumpulan dan Persiapan Data
   Langkah awal yang perlu dilakukan adalah mengumpulkan dan mempersiapkan data sesuai dengan format sehingga dapat digunakan sebagai input untuk algoritma.

2. Pemilihan Fitur
   Data yang sudah diperoleh dari langkah sebelumnya kemungkinan terdiri dari berbagai macam fitur, namun tidak seluruhnya akan relevan terhadap proses *learning*. Fitur yang tidak relevan ini perlu dihapus.

3. Pemilihan Algoritma
   Langkah selanjutnya adalah memilih algoritma yang sesuai dengan masalah yang ingin diselesaikan agar mendapatkan hasil yang terbaik.

4. Pemilihan Model dan Parameter
   Kebanyakan dari algoritma ML memerlukan campur tangan secara manual agar dapat menentukan nilai yang paling tepat untuk setiap parameter.

5. *Training*
   Setelah menentukan algoritma ML dan parameter yang sesuai, lakukan *training* atau latihan menggunakan sebagian dari keseluruhan data sebagai data latihan.

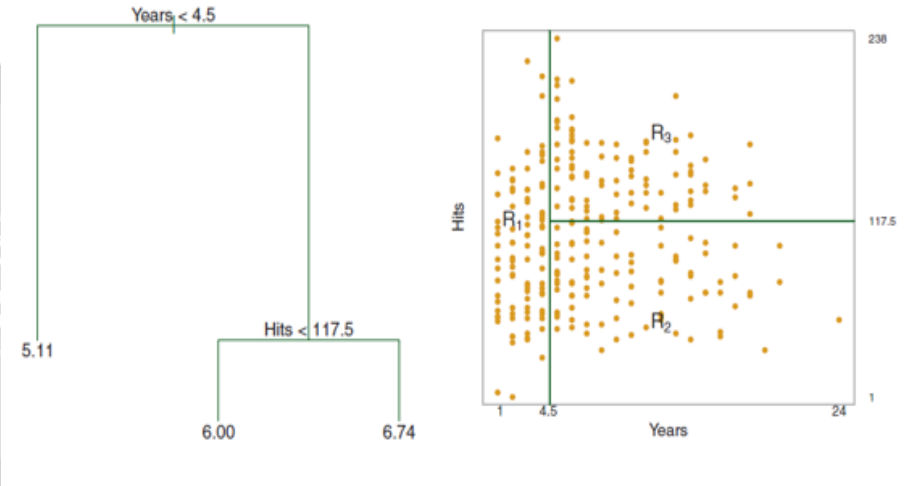
6. Evaluasi Performa
   Langkah terakhir adalah melakukan *testing* atau pengujian model menggunakan data tes untuk mengevaluasi performa model seperti akurasi, presisi, atau *recall*.

2.2 *Tree-Based Models*

2.2.1 Pengertian *Tree-Based Models*

Tree-based models adalah salah satu metode dari algoritma *supervised learning*. Metode ini dapat digunakan untuk masalah klasifikasi maupun regresi. *Tree-based models* untuk masalah klasifikasi dikenal sebagai *Decision Tree* atau pohon keputusan, sedangkan *tree-based models* untuk masalah regresi dikenal

sebagai *Regression Tree* atau pohon regresi. Kedua metode ini adalah metode nonparametrik karena metode ini dapat digunakan dengan mengabaikan asumsi-asumsi yang biasanya harus dipenuhi jika menggunakan metode parametrik. James *et al.* [11] menjelaskan bahwa metode ini melibatkan stratifikasi atau segmentasi ruang prediktor ke beberapa daerah yang sederhana. Untuk membuat prediksi terhadap suatu observasi, biasanya akan digunakan rata-rata atau modus dari data latihan yang sudah ditempatkan ke setiap daerah yang bersesuaian. *Splitting rules* adalah aturan yang digunakan untuk melakukan segmentasi data dan bisa digambarkan dalam bentuk diagram pohon. Proses segmentasi data dan bentuk diagram pohon yang dihasilkan dari *tree-based models* dapat diilustrasikan pada Gambar 2.5.



Gambar 2.5: Proses Segmentasi Data dan Bentuk Diagram Pohon pada *Tree-Based Models*
Sumber: *An Introduction to Statistical Learning* [11]

2.2.2 Terminologi

Beberapa istilah yang biasa digunakan di *tree-based models* adalah sebagai berikut.

1. *Splitting*
   Splitting adalah proses membagi suatu *node* menjadi dua atau lebih *sub-node*.

2. *Branch*
   Branch atau *sub-tree* adalah sub-bagian dari suatu model pohon.

3. *Parent Node* dan *Child Node*

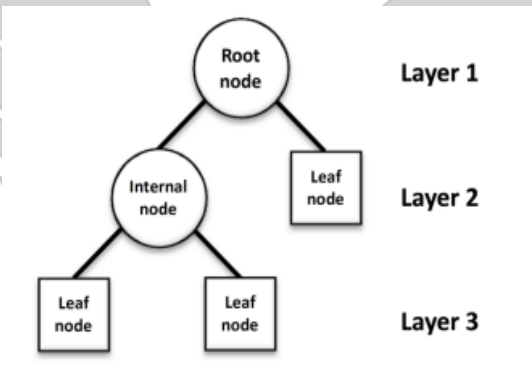
   Node yang berada di bawah *node* lain disebut sebagai *child-node* atau *sub-node*, sedangkan *node* yang mendahului *node* lain disebut sebagai *parent node*.

4. *Depth of Tree*

   Depth of tree atau kedalaman model pohon diukur dari banyaknya *edge* dari *root node* ke suatu *node*. *Edge* adalah jalur yang menghubungkan dua *node*.

2.2.3 Struktur *Tree-Based Models*

Struktur model pohon dimulai dari sebuah *node* yang biasa disebut *root node* dan kemudian akan bercabang (*splitting*) hingga memiliki banyak *leaf node*. Struktur model pohon yang sederhana diilustrasikan pada Gambar 2.6.



Gambar 2.6: Struktur Dasar *Tree-Based Models*
Sumber: *The Use of Facial Micro-Expression State and Tree-Forest Model for Predicting Conceptual-Conflict Based Conceptual Change* [12]

Terdapat tiga jenis *node* sebagai berikut.

1. *Root Node*

   Root node adalah *node* yang berada di pangkal model pohon. *Root node* merepresentasikan seluruh populasi data yang akan dianalisis. *Root node* tidak memiliki *parent node*.

2. *Internal Node*

   Internal node atau *decision node* adalah *node* yang memiliki satu atau lebih *child node*.

3. *Leaf Node*
   Leaf node atau *terminal node* adalah *node* yang tidak melakukan *splitting* sehingga tidak memiliki *child node*.

2.3 *Regression Tree*

Pada penelitian ini, variabel dependen yang digunakan bersifat kontinu, sehingga akan digunakan metode *Regression Tree* untuk merancang model prediksi harga tiket pesawat. Pada *Regression Tree* diasumsikan sebuah model yaitu

$$f(X) = \sum_{m=1}^{M} c_m \cdot 1_{(X \in R_m)} \tag{2.1}$$

atau dapat ditulis juga dalam bentuk

$$f(X) = c_m,\ X \in R_m \tag{2.2}$$

dengan $R_1, ..., R_M$ merepresentasikan partisi dari ruang fitur dan c_m adalah rata-rata dari semua observasi yang ada pada daerah R_m. Indeks m bergerak dari satu hingga M yang artinya jika data X elemen dari suatu partisi R_m, maka hasil estimasi untuk data X adalah rata-rata semua observasi yang ada pada daerah R_m tersebut.

2.3.1 Proses Membangun Model *Regression Tree*

James *et al.* [11] menjelaskan bahwa proses membangun model *Regression Tree* memiliki kurang lebih dua langkah yaitu sebagai berikut.

1. Langkah pertama adalah dengan membagi ruang prediktor, yaitu himpunan nilai untuk $X_1, X_2, ..., X_p$ menjadi sebanyak J daerah yang berbeda dan saling terpisah, $R_1, R_2, ..., R_J$.

2. Untuk setiap observasi yang masuk ke dalam daerah R_j, hasil prediksinya adalah rata-rata variabel respons dari data observasi latihan di daerah R_j.

Sebagai contoh, misalkan pada langkah pertama didapatkan dua daerah R_1 dan R_2, dan rata-rata variabel respons dari data latihan di R_1 adalah sebesar a, sedangkan rata-rata variabel respons dari data latihan di R_2 adalah b. Maka untuk suatu observasi $X = x$, jika $x \in R_1$ akan diprediksi nilai sebesar a, dan jika $x \in R_2$ akan diprediksi nilai sebesar b.

Sekarang akan diuraikan langkah pertama yaitu cara membangun daerah $R_1, ..., R_J$. Secara teori, daerah-daerah tersebut bisa berbentuk apa saja. Namun, untuk mempermudah interpretasi hasil model prediksi, kita akan membagi ruangan prediktor menjadi segi empat berdimensi tinggi atau kotak. Tujuan yang ingin dicapai adalah menemukan kotak-kotak $R_1, ..., R_J$ yang meminimumkan *residual sum of squares* (SSR) yaitu

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2, \tag{2.3}$$

di mana $\hat{y}_{R_j}$ adalah rata-rata variabel respons dari data latihan di kotak ke-j dan y_i adalah nilai aktual dari data ke-i. Nilai eror yaitu selisih antara nilai aktual y_i dan nilai prediksi $\hat{y}_{R_j}$ akan dikuadratkan. Persamaan (2.3) artinya setiap nilai eror kuadrat data ke-i pada kotak R_j akan dijumlahkan. Kemudian indeks j akan bergerak dari satu ke J yang berarti total nilai eror kuadrat tiap kotak $R_1, ..., R_J$ akan dijumlahkan.

Namun, secara komputasi tidak mungkin untuk mempertimbangkan setiap kemungkinan partisi ruang prediktor ke dalam J kotak. Untuk alasan ini, akan digunakan sebuah pendekatan *top-down, greedy* yaitu dengan melakukan *recursive binary splitting*. Pendekatan *top-down* yang berarti dari atas ke bawah karena dilakukan dari puncak pohon, di mana seluruh data observasi masih ada di dalam satu daerah. Setelah itu akan dilakukan *splitting* terhadap daerah prediktor. Setiap *split* ditandai dengan dua *branch* atau cabang yang ada di bawah. Pendekatan ini disebut *greedy* atau serakah karena pada setiap langkah proses pembangunan model pohon, *split* terbaik dibuat di langkah itu sendiri, daripada melihat ke depan dan memilih *split* yang akan menghasilkan model pohon yang lebih baik di beberapa langkah selanjutnya.

Untuk melakukan *recursive binary splitting*, pertama akan dipilih variabel prediktor X_j dan titik potong s sehingga ruang prediktor akan dipecah menjadi daerah $\{X|X_j < s\}$ dan $\{X|X_j \geq s\}$ yang menghasilkan pengurangan terbesar pada nilai RSS. (Notasi $\{X|X_j < s\}$ artinya daerah dari ruang prediktor di mana nilai X_j lebih kecil dari s, sedangkan notasi $\{X|X_j \geq s\}$ artinya daerah dari ruang prediktor di mana nilai X_j lebih besar sama dengan s). Ini artinya, seluruh variabel prediktor $X_1, ..., X_p$ dan setiap titik potong s dari setiap prediktor akan dipertimbangkan dan dipilih yang akan menghasilkan model pohon dengan nilai RSS terendah. Secara lebih rinci, untuk suatu j dan s, definisikan pasangan *half-plane* atau paruh bidang yaitu

$$R_{1(j,\,s)} = \{X|X_j < s\} \text{ dan } R_{2(j,\,s)} = \{X|X_j \geq s\} \tag{2.4}$$

dan akan dicari nilai j dan s yang meminimumkan persamaan

$$\sum_{i:\, x_i \in R_{1(j,\,s)}} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_{2(j,\,s)}} (y_i - \hat{y}_{R_2})^2 \tag{2.5}$$

di mana $\hat{y}_{R_1}$ adalah rata-rata variabel respons untuk data latihan di $R_{1(j,\,s)}$ dan $\hat{y}_{R_2}$ adalah rata-rata variabel respons untuk data latihan di $R_{2(j,\,s)}$. Total nilai eror kuadrat tiap data ke-i yang ada pada tiap ruang $R_{1(j,\,s)}$ dan $R_{2(j,\,s)}$ akan dijumlahkan.

Selanjutnya akan diulangi proses tersebut yaitu mencari variabel prediktor dan titik potong yang terbaik untuk membagi data lebih jauh sehingga nilai RSS dapat diminimumkan untuk setiap daerah. Namun kali ini yang dibagi bukan seluruh ruang prediktor, melainkan salah satu dari dua daerah yang sudah dibagi sebelumnya. Maka sekarang diperoleh tiga daerah. Lalu akan dibagi lagi salah satu dari ketiga daerah tersebut yang meminimumkan RSS. Proses ini akan terus berlanjut hingga suatu kriteria *stopping* sudah dicapai. Misalkan, *splitting* akan dilakukan hingga tidak ada daerah yang memiliki lebih dari lima observasi. Jika daerah-daerah $R_1, ..., R_J$ sudah dibuat, maka prediksi terhadap data tes dapat dilakukan menggunakan rata-rata data latihan di daerah di mana data tes itu berada.

2.4 *The Bias-Variance Trade Off*

Pada bukunya yang berjudul *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch*, Brownlee [13] menjelaskan konsep *trade-off* antara bias dengan varians. Tujuan utama dari suatu algoritma *supervised learning* adalah mendapatkan estimasi *mapping function* atau fungsi pemetaan (F) untuk variabel *output* (Y) terhadap data variabel input (X). Fungsi pemetaan juga dikenal sebagai *target function* atau fungsi sasaran. Menurut Brownlee, eror prediksi untuk suatu algoritma ML dapat dibagi menjadi tiga yaitu sebagai berikut.

1. *Bias Error*

   Bias adalah asumsi penyederhanaan yang dibuat oleh sebuah model agar fungsi sasaran menjadi lebih mudah dimengerti. Pada umumnya, algoritma parametrik memiliki bias yang tinggi sehingga algoritma ini menjadi lebih mudah dimengerti namun tidak terlalu fleksibel. Akibatnya, hasil prediksi

algoritma ini menjadi tidak begitu akurat jika digunakan pada masalah kompleks yang tidak dapat memenuhi asumsi penyederhanaan dari bias algoritma tersebut.

- Bias Rendah: menggunakan lebih sedikit asumsi terhadap bentuk fungsi sasaran.
- Bias Tinggi: menggunakan lebih banyak asumsi terhadap bentuk fungsi sasaran.

Beberapa contoh algoritma ML yang memiliki bias yang rendah adalah *Decision Tree*, *k-Nearest Neighbors*, dan *Support Vector Machine* (SVM). Beberapa contoh algoritma ML yang memiliki bias yang tinggi adalah *Linear Regression*, *Linear Discriminant Analysis*, dan *Logistic Regression*.

2. *Variance Error*

   Varians adalah jumlah dari perubahan estimasi fungsi sasaran jika data latihan yang digunakan berbeda. Fungsi sasaran diestimasi dari data latihan oleh algoritma ML, maka kita harus mengantisipasi bahwa algoritma tersebut akan memiliki beberapa varians. Idealnya, perubahan antara satu set data latihan tidak begitu jauh dengan set data lainnya, yang artinya algoritma tersebut bagus dalam menemukan pemetaan tersembunyi antara variabel input dan variabel *output*. Algoritma ML yang variansnya tinggi sangat dipengaruhi oleh data latihan.

   - Varians Rendah: perubahan kecil pada estimasi fungsi sasaran terhadap perubahan pada data latihan.
   - Varians Tinggi: perubahan besar pada estimasi fungsi sasaran terhadap perubahan pada data latihan.

   Pada umumnya, algoritma ML nonparametrik yang fleksibel memiliki varians yang tinggi. Sebagai contoh, *Decision Tree* memiliki varians yang tinggi, bahkan akan lebih tinggi jika tidak dilakukan *pruning* atau pemangkasan pada model pohon tersebut. Beberapa contoh algoritma ML yang memiliki varians rendah adalah *Linear Regression*, *Linear Discriminant Analysis*, dan *Logistic Regression*. Beberapa contoh algoritma ML yang memiliki varians tinggi adalah *Decision Tree*, *k-Nearest Neighbors*, dan *Support Vector Machine* (SVM).

3. *Irreducible Error*

   Irreducible error adalah eror yang tidak dapat dikurangi terlepas dari algoritma apa yang digunakan. Eror ini bisa ada karena beberapa faktor seperti variabel yang tidak diketahui yang mungkin saja sebenarnya dapat memengaruhi pemetaan variabel input ke variabel *output*.

Tujuan utama dari setiap algoritma ML adalah mencapai bias dan varians yang rendah agar dapat memprediksi dengan tepat. Namun hubungan antara bias dan varians tidak dapat dihindari, yang artinya meningkatkan bias akan menurunkan varians sedangkan meningkatkan varians akan menurunkan bias. Ada *trade-off* atau tarik-ulur antara dua masalah ini. Kita perlu mencari kesetimbangan di hubungan *trade-off* tersebut. Dalam kenyataannya, nilai sebenarnya dari bias dan varians tidak diketahui karena kita tidak tahu fungsi sasaran sebenarnya.

2.5 *Overfitting* dan *Underfitting*

Brownlee [13] menyatakan bahwa penyebab performa yang buruk pada ML adalah antara karena *overfitting* atau *underfitting* data. Brownlee juga menjelaskan bagaimana model yang baik, yaitu model yang tidak *overfit* dan *underfit*.

Proses *learning* fungsi sasaran dari data latihan termasuk dalam *inductive learning*. *Induction* atau induktif mengacu pada pembelajaran konsep-konsep umum. Masalah inilah yang ingin dipecahkan *supervised learning*. Generalisasi mengacu pada seberapa baik konsep-konsep yang dipelajari sebuah ML diterapkan ke contoh tertentu yang tidak ditangkap oleh model tersebut saat melakukan *learning*. Tujuan yang ingin dicapai oleh suatu model ML yang baik adalah menggeneralisasi dengan baik dari data latihan ke data apa pun, sehingga model ini dapat melakukan prediksi di masa mendatang pada data yang belum pernah ditemui oleh model tersebut.

2.5.1 *Overfitting*

Overfitting mengacu pada sebuah model yang memodelkan data latihan dengan terlalu baik. *Overfitting* terjadi saat suatu model mempelajari detil dan *noise* di data latihan hingga berdampak negatif pada performa model terhadap data baru. *Noise* adalah data yang nilainya salah atau merupakan sebuah anomali. Ini berarti *noise* atau fluktuasi acak di data latihan diambil dan dipelajari sebagai

konsep oleh model tersebut, padahal konsep tersebut tidak dapat diaplikasikan ke data baru. Ini memberi dampak negatif terhadap kemampuan model untuk menggeneralisasi.

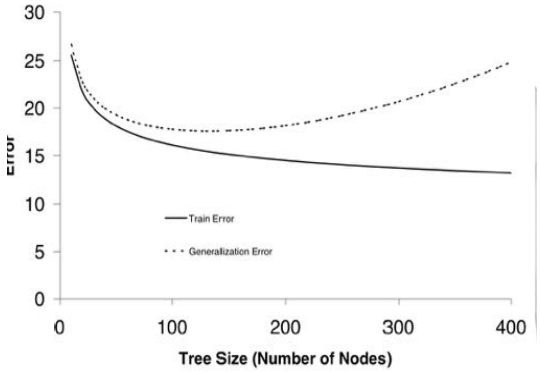
Overfitting lebih mungkin terjadi pada model nonparametrik dan nonlinear yang memiliki fleksibilitas saat melakukan *learning*, sehingga banyak algoritma nonparametrik yang menambahkan parameter atau teknik untuk membatasi seberapa detil suatu model saat melakukan *learning*. Sebagai contoh, *Decision Tree* adalah algoritma nonparametrik yang fleksibel dan rawan terjadi *overfitting* data latihan. Masalah ini dapat diatasi dengan melakukan *pruning*.

2.5.2 *Underfitting*

Underfitting mengacu pada sebuah model tidak mampu memodelkan data latihan dan tidak dapat melakukan generalisasi terhadap data baru. Model yang *underfit* bukan merupakan model yang bagus dan jelas akan memiliki performa yang buruk pada data latihan. Solusi dari masalah ini adalah mencoba model ML alternatif.

2.5.3 *Fit* yang Baik pada *Machine Learning*

Tujuan yang ingin dicapai adalah mendapatkan model yang ideal yaitu model yang berada di posisi terbaik antara *underfitting* dan *overfitting*. Untuk memahami tujuan ini, dapat dilihat performa sebuah algoritma ML seiring algoritma tersebut melakukan *learning* pada data latihan. Hal ini dapat dilakukan dengan membuat *plot* yang menggambarkan kemampuan model pada data latihan dan data tes seperti yang diilustrasikan pada Gambar 2.7.



Gambar 2.7: *Overfitting* pada *Decision Trees*
Sumber: *Using Decision Tree-Based Data Mining to Predict Types of Apparels* [14]

Seiring model tersebut melakukan *learning*, eror model pada data latihan menurun dan sama halnya dengan eror pada data tes. Jika latihan dilakukan terlalu lama, performa terhadap data latihan tetap terus menurun karena model tersebut *overfitting* dan sedang mempelajari detil yang tidak penting dan *noise* dari data latihan. Dalam waktu yang bersamaan, eror terhadap data tes mulai meningkat karena kemampuan model untuk melakukan generalisasi mulai memburuk. Posisi yang terbaik adalah saat sebelum *training error* mulai meningkat kembali, yaitu ketika model memiliki kemampuan yang baik pada memodelkan data latihan dan menggeneralisasi data tes.

2.6 Meningkatkan Akurasi *Regression Tree*

Untuk meningkatkan akurasi pada model pohon regresi, dapat dilakukan ketiga metode berikut.

2.6.1 *Pruning*

James *et al.* [11] menjelaskan proses melakukan *pruning* pada *Regression Tree*. Proses pembuatan model *Regression Tree* yang dijelaskan pada bagian 2.3.1 dapat menghasilkan prediksi yang baik pada data latihan, tetapi akan cenderung terjadi *overfitting* sehingga hasil prediksi pada data tes akan menjadi buruk. Ini dikarenakan model pohon yang dihasilkan terlalu kompleks. Model pohon dengan jumlah *split* yang lebih sedikit (yang artinya lebih sedikit daerah $R_1, ..., R_J$) dapat menghasilkan varians yang lebih rendah dengan mengorbankan sedikit bias.

Strategi yang dapat dilakukan adalah dengan membangun *tree* T_0 yang sangat besar, lalu lakukan *pruning* atau pemangkasan agar mendapatkan sebuah *sub-tree*. Tujuan yang ingin dicapai adalah memilih sebuah *sub-tree* yang menghasilkan tingkat eror terendah. Untuk menentukan *sub-tree* yang terbaik, akan sangat sulit jika harus mempertimbangkan seluruh kemungkinan *sub-tree* karena jumlahnya akan sangat besar. Maka dari itu, dapat digunakan *cost complexity pruning* yang dikenal juga sebagai *weakest link pruning*, sehingga hanya perlu mempertimbangkan urutan *tree* yang diindeks dengan *nonnegative tuning parameter* α. *Tuning parameter* akan mengatur *trade-off* antara kompleksitas *tree* dengan akurasinya. Algoritma *pruning Regression Tree* adalah sebagai berikut.

1. Lakukan *recursive binary splitting* untuk membangun sebuah *tree* yang besar menggunakan data latihan, dan berhenti ketika setiap *terminal node* sudah tidak memenuhi syarat jumlah minimum observasi.

2. Terapkan *cost complexity pruning* ke *tree* yang besar tersebut untuk mendapatkan sebuah urutan dari *sub-tree* yang terbaik, sebagai fungsi dari α.

3. Lakukan *K-fold cross-validation* untuk memilih α, yaitu dengan membagi data latihan menjadi sebanyak *K fold*. Untuk setiap *fold* dari $h = 1, \ldots, K$ akan dilakukan dua langkah berikut.

   a. Ulangi langkah pertama dan kedua untuk semua *fold* kecuali *fold* ke-h.

   b. Evaluasi *mean squared prediction error* pada data di fold ke-h, sebagai fungsi dari α.

   Kemudian rata-ratakan nilai dari setiap α, dan pilih α yang meminimumkan eror rata-rata.

4. Pilih *sub-tree* dari langkah kedua sesuai dengan nilai yang dipilih dari α.

Untuk setiap nilai dari α, ada *sub-tree* sehingga $T \subset T_0$ sehingga

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \tag{2.6}$$

menjadi sekecil mungkin. $|T|$ adalah jumlah *terminal node* dari *tree* T, R_m adalah *subset* dari ruang prediktor yaitu *terminal node* ke-m, dan $\hat{y}_{R_m}$ adalah prediksi

respons dari R_m, yaitu rata-rata dari data observasi di R_m. Setiap nilai eror kuadrat dan $\alpha|T|$ dari data ke-i yang ada pada ruang R_m akan dijumlahkan. Indeks m bergerak dari satu hingga $|T|$ yang artinya total nilai eror kuadrat dan $\alpha|T|$ pada tiap ruang $R_1, ..., R_{|T|}$ akan dijumlahkan. Jika $\alpha = 0$, maka *sub-tree* T sama dengan T_0, karena Persamaan (2.6) menjadi hanya menghitung *training error*. Namun seiring nilai α meningkat, ada nilai yang harus dibayarkan kalau memilih *tree* dengan banyak *terminal nodes*, sehingga nilai dari Persamaan (2.6) cenderung akan mengecil untuk *sub-tree* yang lebih kecil.

2.6.2 *Bagging*

James *et al.* [11] menjelaskan konsep metode *Bagging Regression Tree*. Seperti yang dijelaskan pada bagian 2.4 mengenai *trade-off* antara bias dan varians, model *Regression Tree* cenderung memiliki varians yang tinggi. Ini artinya jika data latihan dibagi menjadi dua dan dibuat dua model *Regression Tree* dari kedua data tersebut, hasil model yang didapatkan bisa sangat berbeda. *Bootstrap aggregation* atau *bagging* adalah prosedur yang dilakukan untuk mengurangi varians dari sebuah model pohon.

Untuk mengurangi varians dan meningkatkan akurasi prediksi dari model *Regression Tree*, dapat dilakukan banyak pengambilan data latihan dari sebuah populasi lalu buat model prediksi untuk masing-masing data latihan. Setelah itu, rata-ratakan hasil prediksi yang didapatkan. Dalam kata lain, nilai $\hat{f}^1(x), \hat{f}^2(x), ..., \hat{f}^B(x)$ dapat dihitung menggunakan B set data latihan yang berbeda, dan dirata-ratakan untuk mendapatkan sebuah model *learning* dengan varians rendah yaitu,

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x). \tag{2.7}$$

Tentunya ini bukan cara yang praktis karena pada kenyataannya, sulit untuk mendapatkan banyak data latihan. Oleh karena itu, dapat dilakukan *bootstraping* yaitu dengan mengambil ulang sampel dari sebuah data latihan. Melalui pendekatan ini, akan dihasilkan B set data latihan *bootstrap* yang berbeda. Lalu lakukan *training* metode pada set data *bootstrap* ke-b untuk mendapatkan $\hat{f}^{*b}(x)$ dan rata-ratakan semua hasil prediksi sehingga akan diperoleh

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x). \tag{2.8}$$

Model *Bagging Regression Tree* biasa dibuat tanpa adanya *pruning* sehingga model pohonnya biasa besar. Oleh karena itu, masing-masing *tree* mempunyai varians yang besar, namun biasnya rendah. Oleh karena itu, merata-ratakan setiap *tree* akan mengurangi varians.

Seperti yang kita ketahui, *bagging* biasanya akan meningkatkan akurasi prediksi dibandingkan jika hanya menggunakan satu *tree*. Namun akan sulit untuk menginterpretasikan model tersebut karena jumlahnya yang banyak, sehingga variabel-variabel yang penting menjadi kurang jelas. Oleh karena itu, *bagging* meningkatkan akurasi prediksi dengan mengorbankan kemudahan dalam interpretasi.

Walaupun kumpulan *bagged trees* lebih sulit untuk diinterpretasi dibandingkan dengan satu *tree*, kita masih dapat memperoleh rangkuman secara keseluruhan mengenai tingkat kepentingan setiap variabel prediktor melalui RSS seperti pada Persamaan (2.3). Hal ini dapat dilakukan dengan mengurutkan jumlah total pengurangan RSS yang disebabkan oleh *splitting* dari setiap prediktor, kemudian dirata-ratakan pada sebanyak B pohon. Nilai yang besar menandakan bahwa prediktor tersebut penting.

2.6.3 *Random Forest*

James *et al.* [11] menjelaskan bahwa metode *Random Forest* bisa memberikan sedikit perbaikan dari *bagged trees* dengan mengurangi korelasi antar *tree*. *Bagging Regression Tree* membangun sejumlah *tree* dari data latihan *bootstrap* dan mempertimbangkan seluruh prediktor saat menentukan *split*. Namun pada *Random Forest*, sebanyak m prediktor dari seluruh p predikor akan disampel secara acak sebagai kandidat *split*. *Split* akan ditentukan hanya dari m prediktor tersebut. Sampel m prediktor baru akan diambil lagi pada setiap *split*. Nilai m yang digunakan biasanya adalah $m \approx \sqrt{p}$. Dalam kata lain, jumlah prediktor yang dipertimbangkan pada setiap *split* sekitar akar dari jumlah total prediktor.

Ada alasan penting dibalik metode sampel acak prediktor tersebut. Misalkan ada satu prediktor yang sangat kuat dari sebuah set data dengan beberapa prediktor lainnya yang cukup kuat. Maka pada kumpulan *bagged trees*, sebagian besar atau semua *tree* akan menggunakan prediktor yang sangat kuat tersebut sebagai *split*

yang paling atas. Akibatnya, semua *bagged trees* akan saling menyerupai dan hasil prediksinya akan saling berkorelasi. Merata-ratakan *tree* yang saling berkorelasi tidak akan mengurangi varians sebanyak *tree* yang saling tidak berkorelasi. Jika model *Random Forest* dibangun menggunakan $m = p$ prediktor, maka metode ini sama saja dengan *Bagging Regression Tree*.

2.6.4 *Hyperparameter Optimization*

Elshawi *et al.* [15] menjelaskan bahwa secara umum, membangun model ML yang efektif adalah proses yang kompleks dan memakan banyak waktu. Proses ini melibatkan penentuan algoritma yang sesuai dan mendapatkan arsitektur model yang optimal dengan melakukan *hyperparameter* (HP) *tuning*. Menurut Kuhn & Johnson [16], terdapat dua jenis parameter, yaitu parameter model yang dapat diinisialisasi dan diperbarui melalui proses pembelajaran data dan *hyperparameter* yang tidak dapat diestimasi langsung saat proses pembelajaran data dan harus ditentukan sebelum melakukan *training* model ML karena parameter ini mendefinisikan arsitektur model. Hutter *et al.* [17] menjelaskan bahwa untuk membangun model ML yang optimal, ada banyak kemungkinan yang harus dieksplorasi. Proses merancang arsitektur model yang ideal dengan konfigurasi HP yang optimal disebut dengan *hyperparameter tuning*. Proses optimisasi ini dianggap sebagai kunci membangun model ML yang efektif, terutama untuk *tree-based models* dan *deep neural networks*, yang memiliki banyak HP. Sanders & Giraud-Carrier [18] menyatakan bahwa pada model berbasis pohon, proses optimisasi dapat dilakukan pada banyak HP seperti maksimum tingkat kedalaman model pohon, maksimum jumlah fitur, minimum jumlah data yang diperlukan untuk melakukan *splitting*, minimum jumlah data dalam sebuah *terminal node*, maksimum jumlah *terminal node*, dan lain-lain.

2.7 *k-Fold Cross-Validation*

James *et al.* [11] menjelaskan konsep *k-fold cross-validation* (CV). *Resampling method* berkaitan dengan pengambilan sampel berulang kali dari sebuah set data latihan dan perancangan model dari tiap sampel untuk memperoleh informasi tambahan terhadap sebuah model. Salah satu *resampling method* yang sering digunakan adalah *k-fold cross-validation*. Metode ini dilakukan dengan membagi sebuah set data observasi menjadi k kelompok atau *folds* yang berukuran

kurang lebih sama. Kelompok pertama diperlakukan sebagai set data validasi, dan model akan dibuat dengan sisa $k-1$ kelompok tersebut. Kemudian *Mean Squared Error* (MSE) akan dihitung menggunakan data validasi. MSE adalah rata-rata nilai kuadrat dari selisih antara nilai aktual dengan nilai prediksi. Persamaan untuk MSE adalah

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.9}$$

dengan y_i merupakan nilai aktual dari data ke-i, $\hat{y}_i$ merupakan nilai prediksi dari y_i, dan n merupakan jumlah total data.

Prosedur tersebut akan diulangi sebanyak k kali. Setiap kalinya, kelompok data observasi yang berbeda akan diperlakukan sebagai set data validasi. Proses ini akan menghasilkan sebanyak k estimasi eror tes, $MSE_1, MSE_2, ..., MSE_k$. Estimasi dari *k-fold* CV dihitung dengan merata-ratakan nilai MSE dari setiap kelompok atau *fold* tersebut,

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i, \tag{2.10}$$

dengan k sebagai total jumlah kelompok atau *fold*. Nilai k yang dipilih biasanya adalah $k = 5$ atau $k = 10$.

2.8 Evaluasi Model

Mayer & Butler [19] menyatakan bahwa untuk mengevaluasi performa suatu model, dapat dilakukan perhitungan penyimpangan berdasarkan selisih nilai sebenarnya dengan nilai prediksi. Perhitungan eror yang dapat dilakukan seperti *Mean Absolute Error*, *Mean Absolute Percentage Error*, dan *Root Mean Squared Error*. Performa model juga dapat dievaluasi melalui besar varians.

2.8.1 *Root Mean Squared Error*

Chai *et al.* [20] menjelaskan bahwa dalam perhitungan *Root Mean Squared Error* (RMSE), nilai eror yang besar akan diberikan bobot yang relatif lebih besar. Ini karena nilai eror dikuadratkan terlebih dahulu sebelum dirata-ratakan, sehingga RMSE adalah akar dari rata-rata nilai kuadrat dari selisih antara nilai aktual dengan nilai prediksi. Persamaan untuk RMSE adalah

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{2.11}$$

dengan y_i merupakan nilai aktual dari data ke-i, $\hat{y}_i$ merupakan nilai prediksi dari y_i, dan n merupakan jumlah total data.

2.8.2 *Mean Absolute Error*

Chai *et al.* [20] menyatakan bahwa selain RMSE, *Mean Absolute Error* (MAE) merupakan salah satu perhitungan yang sering digunakan dalam evaluasi model. Nilai dari MAE cenderung lebih kecil dari RMSE karena MAE memberikan bobot yang sama rata untuk setiap eror. MAE adalah rata-rata nilai mutlak dari selisih antara nilai aktual dengan nilai prediksi. Persamaan untuk MAE adalah

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{2.12}$$

dengan y_i merupakan nilai aktual dari data ke-i, $\hat{y}_i$ merupakan nilai prediksi dari y_i, dan n merupakan jumlah total data.

2.8.3 *Mean Absolute Percentage Error*

De Myttenaere [21] menyatakan bahwa perhitungan *Mean Absolute Percentage Error* (MAPE) sering digunakan karena interpretasinya yang sangat intuitif menggunakan eror relatif. Perhitungan MAPE juga menggunakan nilai mutlak dari selisih antara nilai aktual dengan nilai prediksi. Nilai mutlak eror tersebut kemudian akan dibagi dengan nilai aktual sehingga didapatkan nilai eror relatif. Nilai eror relatif akan dikali 100% untuk mendapatkan persentase eror sehingga persamaan untuk MAPE adalah

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{\hat{y}_i}\right| \times 100\%, \tag{2.13}$$

dengan y_i merupakan nilai aktual dari data ke-i, $\hat{y}_i$ merupakan nilai prediksi dari y_i, dan n merupakan jumlah total data. Interpretasi nilai MAPE dijelaskan oleh Lewis [22] pada Tabel 2.1.

Tabel 2.1: Interpretasi Nilai MAPE

MAPE	Interpretasi
<10%	Akurasi prediksi sangat baik
10-20%	Akurasi prediksi baik
20-50%	Akurasi prediksi layak
>50%	Akurasi prediksi buruk

2.8.4 Koefisien Determinasi

Di Bucchianico [23] menjelaskan bahwa koefisien determinasi atau R^2 adalah sebuah konsep dalam analisis varians dan analisis regresi, yaitu sebuah ukuran proporsi varians yang dapat dijelaskan dalam data. Oleh karena itu, semakin besar nilai R^2, semakin baik sebuah model menjelaskan data. Ketika data terdiri dari variabel respons $y_1, ..., y_n$ dan sebuah model dengan variabel prediktor diterapkan pada data, maka R^2 dapat didefinisikan sebagai

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}, \tag{2.14}$$

dengan $\bar{y}$ merupakan rata-rata dari observasi, y_i merupakan nilai aktual dari data ke-i, $\hat{y}_i$ merupakan nilai prediksi dari y_i, dan n merupakan jumlah total data. Kisaran nilai dari R^2 adalah antara 0 dan 1.

2.9 Tinjauan Pustaka

Beberapa referensi yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Penelitian yang dilakukan oleh K. Tziridis, Th. Kalampokas, dan G. A. Papakostas dengan judul *Airfare Prices Prediction Using Machine Learning Techniques* [24]. Penelitian ini menggunakan delapan metode ML untuk memprediksi harga tiket pesawat. Metode-metode yang digunakan adalah *Multilayer Perceptron* (MLP), *Generalized Regression Neural Network*, *Extreme Learning Machine* (ELM), *Random Forest Regression Tree*, *Regression Tree*, *Bagging Regression Tree*, *Regression SVM (Polynomial and Linear)*, dan *Linear Regression* (LR). Data yang digunakan adalah data penerbangan dari bulan Desember sampai bulan Juli untuk rute

Thessaloniki-Stuttgart. Untuk setiap data penerbangan, akan diambil delapan fitur dari sebuah *web site* yaitu waktu keberangkatan, waktu kepergian, jumlah bagasi yang gratis, jumlah hari sebelum keberangkatan, jumlah transit, hari libur, penerbangan semalam, dan hari penerbangan. Hasil penelitian pada artikel ini menunjukkan bahwa *Bagging Regression Tree*, *Random Forest Regression Tree*, *Regression Tree*, dan MLP adalah model yang memiliki performa paling baik dalam memprediksi harga tiket pesawat. Hasil penelitian juga menunjukkan bahwa waktu keberangkatan dan waktu kedatangan memengaruhi harga tiket pesawat secara signifikan.

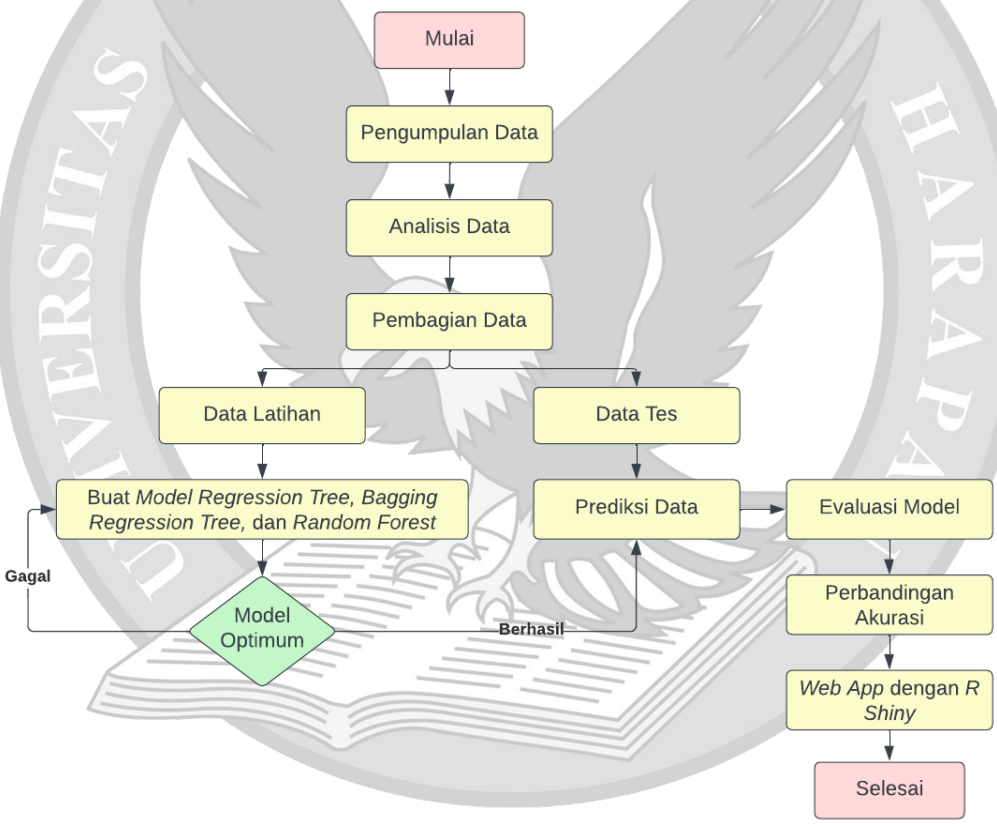
2. Artikel yang dilakukan oleh Carolin Strobl, James Malley, dan Gerhard Tutz [25] dengan judul *An Introduction to Recursive Partitioning: Rationale, Applicationand Characteristics of Classification and Regression Trees, Bagging and Random Forests*. Artikel ini membahas konsep-konsep penting pada metode *Decision Tree*, *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*.

3. Penelitian yang dilakukan oleh Camino González, José Mira-McWilliams, dan Isabel Juárez [26] dengan judul *Important variable assessment and electricity price forecasting based on Regression Tree models: Classification and Regression Trees, Bagging and Random Forests*. Penelitian ini menggunakan metode *Classification and Regression Trees* (CART), *Bagging Regression Tree*, dan *Random Forests* untuk memprediksi harga biaya listrik di *Spanish-Iberian Market*. Data yang digunakan membangun model diambil dari 1 Januari 2000 hingga 31 Desember 2011. Ada 22 variabel prediktor yang digunakan pada penelitian ini yaitu hari, bulan, jam, nuklir, permintaan (dalam jam), dan sebagainya. Penelitian ini mengidentifikasi variabel prediktor mana yang paling memengaruhi biaya listrik melalui konsep *Variable Importance*, dan didapatkan bahwa variabel jam sebagai variabel yang paling penting. Penelitian ini juga menghitung akurasi tiap model menggunakan MAPE dan diperoleh hasil bahwa *Random Forest* adalah metode yang paling akurat.

BAB III

METODOLOGI

Pada bab ini akan dijelaskan data yang akan digunakan dalam penelitian ini. Selain itu akan dijelaskan pula langkah-langkah untuk mencapai tujuan pada penelitian ini, yaitu membangun model prediksi harga tiket pesawat di Indonesia menggunakan *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*, kemudian membandingkan keakuratan ketiga metode tersebut. Data yang digunakan dalam penelitian ini adalah data penerbangan dari 9 Juni 2022 hingga 31 Agustus 2022. Langkah-langkah yang akan dilakukan dalam penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1: *Flowchart* Langkah-Langkah Pengerjaan

3.1 Pengumpulan Data

Data yang akan digunakan dalam penelitian ini diperoleh dari sebuah situs *web* penyedia jasa pemesanan tiket pesawat di Indonesia yaitu *tiket.com*. Data penerbangan yang akan diamati adalah data yang berada di antara 9 Juni 2022 hingga 31 Agustus 2022. Rute yang digunakan adalah dua domestik yaitu Jakarta-Surabaya (CGK-SUB) dan Jakarta-Manado (CGK-MDC). Data tersebut akan dikumpulkan setiap harinya menggunakan *browser extension* yaitu *Web Scraper*. Data mentah yang didapatkan dari proses *scraping* ditunjukkan pada Tabel 3.1.

Tabel 3.1: Data Mentah

date	*airline-name*	*departure-time*	*arrival-time*	*transit*	*price*
Min, 7 Agt	Batik Air	04:00	05:30	Langsung	IDR 927.200
Sab, 30 Jul	Super Air Jet	14:45	16:15	Langsung	IDR 816.100
Kam, 14 Jul	Citilink	05:30	07:00	Langsung	IDR 870.451
Kam, 14 Jul	Citilink	11:55	13:25	Langsung	IDR 925.951
Rab, 6 Jul	Lion Air	17:30	19:00	Langsung	IDR 801.100

Data yang telah diperoleh akan dibersihkan sesuai dengan struktur yang sudah ditentukan. Proses *scraping* terkadang menghasilkan *missing values*. Data yang memiliki *missing values* tidak akan diikutsertakan dalam penelitian ini. Beberapa contoh data yang telah dibersihkan ditunjukkan pada Tabel 3.2. Data penerbangan lebih lengkap untuk rute Jakarta-Surabaya dan Jakarta-Manado terlampir pada Lampiran A.

Tabel 3.2: Data Bersih

airline	departure	arrival	transit	day_of_ week	holiday	days_ left_ until_ departure	duration (jam)	scrape_day	scrape_holiday	price (rupiah)
Batik Air	4:00	5:30	0	Sun	TRUE	59	1,5	5	FALSE	927200
Super Air Jet	14:45	16:15	0	Sat	TRUE	51	1,5	5	FALSE	816100
Citilink	5:30	7:00	0	Thu	FALSE	35	1,5	5	FALSE	870451
Citilink	11:55	13:25	0	Thu	FALSE	35	1,5	5	FALSE	925951
Lion Air	17:30	19:00	0	Wed	FALSE	27	1,5	5	FALSE	801100

Variabel-variabel pada data yang ditunjukkan di Tabel 3.2 adalah:

1. variabel '*airline*', menunjukkan maskapai penerbangan di mana maskapai penerbangan yang digunakan dalam penelitian ini adalah Batik Air, Citilink, Garuda Indonesia, Lion Air, Nam Air, dan Super Air Jet,
2. variabel '*departure*', menunjukkan jam keberangkatan,
3. variabel '*arrival*', menunjukkan jam kedatangan,
4. variabel '*transit*', menunjukkan jumlah transit yang akan dilakukan di mana jumlah transit yang mungkin dilakukan berkisar di antara nol sampai empat,
5. variabel '*day_of_week*', menunjukkan hari penerbangan seperti Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, dan Minggu,
6. variabel '*holiday*', menunjukkan hari libur pada hari penerbangan di mana data *TRUE* menunjukkan hari libur dan data *FALSE* menunjukkan hari tidak libur,
7. variabel '*days_left_until_departure*', menunjukkan jumlah hari sebelum hari keberangkatan,
8. variabel '*duration*', menunjukkan durasi penerbangan,
9. variabel '*scrape_day*', menunjukkan hari sebuah data didapatkan seperti Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, dan Minggu,
10. variabel '*scrape_holiday*', menunjukkan hari libur pada hari sebuah data didapatkan di mana data *TRUE* menunjukkan hari libur dan data *FALSE* menunjukkan hari tidak libur, dan
11. variabel '*price*', menunjukkan harga tiket pesawat.

Tabel 3.3: Variabel pada Data Penerbangan

Variabel	Nama Variabel
x_1	airline
x_2	departure
x_3	arrival
x_4	transit
x_5	day_of_week
x_6	holiday
x_7	days_left_until_departure
x_8	duration
x_9	scrape_day
x_{10}	scrape_holiday
y	price

Pada Tabel 3.3 diperlihatkan variabel-variabel yang digunakan pada penelitian ini. Variabel dependen pada penelitian ini adalah harga tiket pesawat yaitu variabel '*price*'. Variabel independen pada penelitian ini adalah variabel '*airline*', '*departure*', '*arrival*', '*transit*', '*day_of_week*', '*holiday*', '*days_left_until_departure*', '*duration*', '*scrape_day*', dan '*scrape_holiday*'.

3.2 Analisis Data

Pada data yang sudah dikumpulkan pada langkah sebelumnya, akan dilakukan beberapa analisis sederhana. Analisis data yang akan dilakukan adalah dengan memvisualisasikan harga tiket pesawat berdasarkan setiap variabel prediktor. Visualisasi data berupa grafik garis dan dibuat menggunakan tabel *pivot*.

3.3 Pembagian Data

Data yang telah dikumpulkan akan dibagi menjadi dua, yaitu data latihan dan data tes. Proporsi pembagian data adalah 80% untuk data latihan dan 20% untuk data tes. Pemilihan data akan dilakukan secara acak. Perancangan model akan menggunakan data latihan dan diuji menggunakan data tes.

3.4 Model Prediksi dengan *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*

Data latihan akan digunakan untuk membangun tiga model prediksi menggunakan *tree-based models*, yaitu *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*. Model *Regression Tree* akan dibangun menggunakan *package rpart*. Model *Bagging Regression Tree* dan *Random Forest* akan dibangun menggunakan *package caret*. Model prediksi tersebut akan menghasilkan *output* numerik yaitu harga tiket pesawat. Model yang akan digunakan adalah model yang paling optimum. Nilai parameter yang digunakan untuk mendapatkan model yang paling optimum akan dicari menggunakan *k-fold cross-validation*.

Proses pembuatan sebuah *tree* adalah sebagai berikut. Pada setiap *node*, akan dicari variabel prediktor $x_1, x_2, ..., x_{10}$ dengan titik *splitting* yang terbaik. Misalkan terdapat tiga data penerbangan yang dapat dilihat pada Tabel 3.4. Pertama, data awal yang terdiri dari keseluruhan data akan dibagi sesuai dengan titik *splitting* pada setiap variabel prediktor $x_1, x_2, ..., x_{10}$ sehingga data awal akan terbagi menjadi dua.

Tabel 3.4: Contoh Data Penerbangan

data	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y
1	a	0,5	0,6	0	1	TRUE	20	1	5	TRUE	10
2	b	0,6	0,8	1	4	FALSE	2	2	2	FALSE	20
3	c	0,7	0,9	0	5	FALSE	59	3	7	FALSE	50

Misal, pertama akan dipilih variabel x_3 dengan titik *splitting* $x_3 < 0,7$ dan $x_3 \geq 0,7$ sehingga data awal akan terbagi menjadi dua yaitu bagian pertama yang terdiri dari data ke-1 dan bagian kedua yang terdiri dari data ke-2 dan ke-3 seperti yang ditunjukkan pada Tabel 3.5 dan 3.6. Pada setiap bagian, akan dicari rata-rata dari nilai y sehingga didapatkan nilai prediksi $\widehat{y}$ untuk setiap bagian.

Tabel 3.5: Contoh Pembagian Data Bagian Pertama

data	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y	$\widehat{y}$
1	a	0,5	0,6	0	1	TRUE	20	1	5	TRUE	10	10

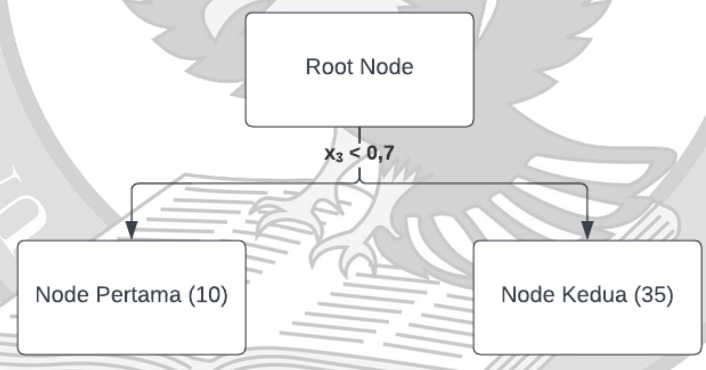
Tabel 3.6: Contoh Pembagian Data Bagian Kedua

data	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y	$\widehat{y}$
2	b	0,6	0,8	1	4	FALSE	2	2	2	FALSE	20	35
3	c	0,7	0,9	0	5	FALSE	59	3	7	FALSE	50	35

Nilai SSR untuk *splitting* ini kemudian didapatkan dengan menggunakan Persamaan (2.3) yaitu

$$SSR = (10-10)^2 + (20-35)^2 + (50-35)^2 = 450 \tag{3.1}$$

sehingga didapatkan nilai SSR untuk variabel x_3 dengan titik *splitting* $x_3 < 0,7$ dan $x_3 \geq 0,7$ sebesar 450. Setelah itu, langkah ini akan diulangi untuk seluruh kemungkinan variabel prediktor dan titik *splitting* lainnya dan nilai SSR-nya akan dibandingkan. Variabel prediktor beserta titik *splitting* dengan nilai SSR terkecil akan menjadi *splitting* pertama. Misalkan dari seluruh nilai SSR yang didapatkan, variabel prediktor x_3 dengan titik *splitting* $x_3 < 0,7$ dan $x_3 \geq 0,7$ memiliki nilai SSR terkecil, maka variabel beserta titik *splitting* tersebut akan menjadi *splitting* pertama seperti yang ditunjukkan pada Gambar 3.2.



Gambar 3.2: Contoh *Splitting* Pertama

Data awal yang terdiri dari seluruh data penerbangan pada *root node* akan terbagi menjadi dua di *node* pertama dan *node* kedua. Untuk mencari *splitting* selanjutnya, akan diulangi kembali langkah sebelumnya yaitu data pada *node* pertama dan *node* kedua akan dibagi sesuai titik *splitting* pada setiap variabel prediktor $x_1, x_2, ..., x_{10}$ dan akan dipilih berdasarkan nilai SSR-nya.

3.5 Memprediksi Data

Jika model *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* yang paling optimum sudah didapatkan, maka ketiga model tersebut sudah siap digunakan untuk melakukan prediksi terhadap data tes yaitu 20% dari keseluruhan data, sehingga data prediksi adalah data penerbangan selama tiga bulan dari 9 Juni 2022 hingga 31 Agustus 2022 dengan rute penerbangan Jakarta-Surabaya (CGK-SUB) dan Jakarta-Manado (CGK-MDC). Prediksi dilakukan menggunakan variabel-variabel seperti jam kedatangan, jam keberangkatan, dan sebagainya, seperti yang dijelaskan pada bagian 3.1. Setiap model akan menghasilkan sebuah estimasi harga tiket pesawat untuk tiap data penerbangan. Harga estimasi tiket pesawat ini nantinya dapat dibandingkan dengan harga aslinya sebagai evaluasi model.

3.6 Evaluasi Model

Berdasarkan hasil estimasi yang didapatkan dari bagian 3.5, setiap model akan dievaluasi performanya dalam memprediksi harga tiket pesawat. Evaluasi model akan dilakukan dengan menghitung besar eror antara data aktual dengan data prediksi menggunakan MAE, RMSE, dan MAPE. Performa model juga diamati menggunakan *goodness-of-fit* yaitu koefisien determinasi dari setiap model. Misal, pada Tabel 3.7 ditunjukkan contoh data penerbangan beserta hasil prediksinya, maka nilai eror prediksi didapatkan dengan menghitung selisih harga tiket pesawat sebenarnya y dengan prediksi harga tiket pesawat $\widehat{y}$. Kemudian nilai eror prediksi tersebut akan digunakan untuk menghitung MAE, RMSE, MAPE, dan koefisien determinasi menggunakan Persamaan (2.11), (2.12), (2.13), dan (2.14).

Tabel 3.7: Contoh Data Penerbangan dan Hasil Prediksinya

data	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y	$\widehat{y}$
1	a	0,5	0,6	0	1	TRUE	20	1	5	TRUE	10	10
2	b	0,6	0,8	1	4	FALSE	2	2	2	FALSE	20	35
3	c	0,7	0,9	0	5	FALSE	59	3	7	FALSE	50	35

Berdasarkan dari hasil evaluasi model, akan ditentukan model yang memiliki performa terbaik dalam memprediksi harga tiket pesawat, yaitu yang memiliki nilai

MAE, RMSE, dan MAPE terkecil. Koefisien determinasi untuk model yang baik adalah yang nilainya besar.

3.7 *Web App* dengan *R Shiny*

Model prediksi harga tiket pesawat akan digunakan untuk membuat sebuah *web app* menggunakan *R Shiny*. Aplikasi ini akan menerima *input* seperti jenis maskapai, jam keberangkatan, jumlah transit, hari keberangkatan, dan sebagainya dan kemudian akan melakukan prediksi harga tiket pesawat menggunakan ketiga model yaitu *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* berdasarkan data yang diterima.

BAB IV

ANALISIS DAN PEMBAHASAN

Pada bab ini dijelaskan hasil dari pengolahan data beserta analisisnya menggunakan metode *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*.

4.1 Data

Data yang telah dikumpulkan menggunakan *Web Scraper* akan dibagi menjadi dua sesuai dengan rutenya. Pada Gambar 4.1 dan Gambar 4.2 ditampilkan lima belas data pertama untuk rute Jakarta-Surabaya dan Jakarta-Manado yang diambil dari *Microsoft Excel*. Kedua data ini akan diolah menggunakan *pivot table* untuk analisis data kemudian diolah menggunakan *R* untuk pembuatan modelnya.

	A	B	C	D	E	F	G	H	I	J	K
1	airline	departure	arrival	transit	day_of_w	holiday	days_left_	duration	scrape_da	scrape_ho	price
2	Batik Air	0.17	0.23	0	1	TRUE	59	1.5	5	FALSE	927200
3	Super Air.	0.61	0.68	0	7	TRUE	51	1.5	5	FALSE	816100
4	Citilink	0.23	0.29	0	5	FALSE	35	1.5	5	FALSE	870451
5	Citilink	0.50	0.56	0	5	FALSE	35	1.5	5	FALSE	925951
6	Lion Air	0.73	0.79	0	4	FALSE	27	1.5	5	FALSE	801100
7	Batik Air	0.38	0.44	0	5	FALSE	21	1.5	5	FALSE	927200
8	Batik Air	0.33	0.40	0	3	FALSE	26	1.5	5	FALSE	992700
9	Super Air.	0.24	0.30	0	2	FALSE	32	1.5	5	FALSE	816100
10	Batik Air	0.21	0.27	0	4	TRUE	69	1.5	5	FALSE	927200
11	Citilink	0.19	0.25	0	6	FALSE	43	1.5	5	FALSE	980341
12	Batik Air	0.29	0.35	0	1	TRUE	59	1.5	5	FALSE	919200
13	Batik Air	0.17	0.23	0	7	FALSE	65	1.5	5	FALSE	927200
14	Super Air.	0.78	0.84	0	6	FALSE	29	1.5	5	FALSE	871600
15	Batik Air	0.25	0.31	0	7	FALSE	65	1.5	5	FALSE	927200
16	Batik Air	0.21	0.27	0	4	FALSE	13	1.5	5	FALSE	862900

Gambar 4.1: Contoh Data Penerbangan Rute Jakarta-Surabaya

	A	B	C	D	E	F	G	H	I	J	K
1	airline	departure	arrival	transit	day_of_w	holiday	days_left_	duration	scrape_da	scrape_ho	price
2	Batik Air	0.11	0.53	1	5	FALSE	63	10.08333	5	FALSE	3409400
3	Batik Air	0.10	0.59	1	4	FALSE	83	11.66667	5	FALSE	2819100
4	Batik Air	0.10	0.41	1	2	FALSE	60	7.416667	5	FALSE	2483900
5	Citilink	0.10	0.25	0	3	FALSE	82	3.416667	5	FALSE	1945544
6	Batik Air	0.11	0.53	1	7	FALSE	44	10.08333	5	FALSE	3409400
7	Lion Air	0.17	0.56	2	2	FALSE	53	9.416667	5	FALSE	3757200
8	Batik Air	0.25	0.40	0	7	FALSE	79	3.5	5	FALSE	1964900
9	Garuda In	0.11	0.26	0	5	FALSE	42	3.416667	5	FALSE	2454900
10	Lion Air	0.08	0.41	1	7	FALSE	2	7.916667	5	FALSE	2255800
11	Lion Air	0.15	0.43	1	5	FALSE	77	6.75	5	FALSE	2249500
12	Citilink	0.10	0.25	0	6	FALSE	64	3.416667	5	FALSE	2190854
13	Lion Air	0.38	0.59	1	4	FALSE	83	5	5	FALSE	2677600
14	Batik Air	0.44	0.59	0	7	FALSE	58	3.5	5	FALSE	2109200
15	Batik Air	0.11	0.53	1	3	FALSE	40	10.08333	5	FALSE	3409400
16	Batik Air	0.08	0.41	1	2	FALSE	18	7.916667	5	FALSE	2549400

Gambar 4.2: Contoh Data Penerbangan Rute Jakarta-Manado

Kolom '*airline*', '*departure*', '*arrival*', '*transit*', '*day_of_week*', '*holiday*', '*days_left_until_departure*', '*duration*', '*scrape_day*', dan '*scrape_holiday*' adalah data untuk variabel prediktor, sedangkan kolom '*price*' adalah data untuk variabel respons. Kolom '*departure*' dan '*arrival*' sebelumnya dalam format *timestamp* namun diubah menjadi numerik. Jam akan diubah dalam bentuk menit dengan mengalikan angka jam dengan 60 dan ditambahkan dengan angka menit, kemudian dibagi dengan total menit dalam sehari yaitu 1440 menit. Misalkan *timestamp* 2.40 akan diubah ke dalam bentuk numerik, maka jam 2 akan dikalikan 60 dan ditambahkan dengan menit 40, kemudian hasilnya akan dibagi dengan 1440 sehingga didapatkan bentuk numerik dari 2.40 yaitu 0,1667. Kolom '*duration*' juga dalam bentuk numerik.

Kolom '*transit*', '*days_left_until_departure*', dan '*price*' dalam bentuk *integer*. Kolom '*holiday*' dan '*scrape_holiday*' dalam bentuk *logical*. Kolom '*airline*', '*day_of_week*', dan '*scrape_day*' diubah menjadi bentuk faktor. Kolom '*airline*' memiliki enam level pada data Jakarta-Surabaya dan empat level pada data Jakarta-Manado, sedangkan kolom '*day_of_week*' dan '*scrape_day*' memiliki tujuh level. Kelas untuk tiap variabel yang digunakan pada penelitian ini beserta beberapa elemen datanya ditunjukkan pada Tabel 4.1.

Tabel 4.1: Kelas dan Elemen untuk Setiap Variabel pada Data Penerbangan

Variabel	Nama Variabel	Kelas	Contoh Elemen Data
x_1	airline	Factor	”Batik Air”, ”Citilink”, ...
x_2	departure	num	0,17, 0,61, 0,23, 0,5, ...
x_3	arrival	num	0,23, 0,68, 0,29, 0,56, ...
x_4	transit	int	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
x_5	day_of_week	Factor	”1”,”2”,”3”,”4”, ...
x_6	holiday	logi	TRUE, TRUE, FALSE, ...
x_7	days_left_until_departure	int	59, 51, 35, 35, ...
x_8	duration	num	1,5, 1,5, 1,5, 1,5, ...
x_9	scrape_day	Factor	”1”,”2”,”3”,”4”, ...
x_{10}	scrape_holiday	logi	FALSE, FALSE, FALSE, ...
y	price	int	927200, 816100, 870451, ...

Pada data penerbangan yang telah dikumpulkan, terdapat *missing values* pada tiap rute. Data dengan *missing values* tidak diikutsertakan dalam penelitian ini. Penjelasan jumlah data untuk tiap rute adalah sebagai berikut.

1. Rute Jakarta-Surabaya memiliki total 103.286 data termasuk 1.889 data dengan *missing values*, sehingga total data bersih adalah 101.397.
2. Rute Jakarta-Manado memiliki total 63.146 data termasuk 8.649 data dengan *missing values*, sehingga total data bersih adalah 54.497.

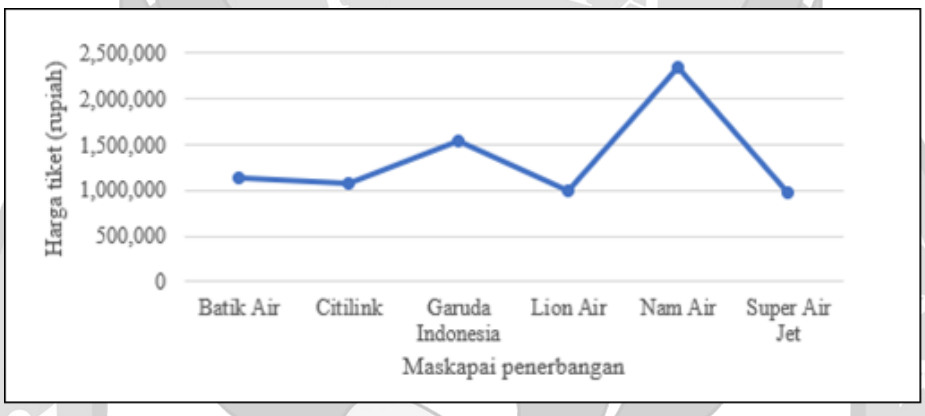
4.2 Analisis Data

Analisis data akan dilakukan menggunakan *pivot table* di *Microsoft Excel*. Harga tiket pesawat (variabel respons) akan divisualisasikan dengan grafik garis berdasarkan 10 faktor (variabel prediktor) yaitu jenis maskapai penerbangan, jam keberangkatan, jam kedatangan, jumlah transit, hari penerbangan, hari libur, jumlah hari tersisa sebelum keberangkatan, durasi penerbangan, hari data penerbangan didapatkan, dan hari libur data penerbangan didapatkan.

4.2.1 Rute Jakarta-Surabaya (CGK-SUB)

4.2.1.1 Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan jenis maskapai penerbangan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.3. Maskapai penerbangan dengan rata-rata harga tiket tertinggi adalah Nam Air, yaitu sebesar Rp2.350.514,00. Maskapai penerbangan dengan rata-rata harga tiket terendah adalah Super Air Jet, yaitu sebesar Rp978.471,00.



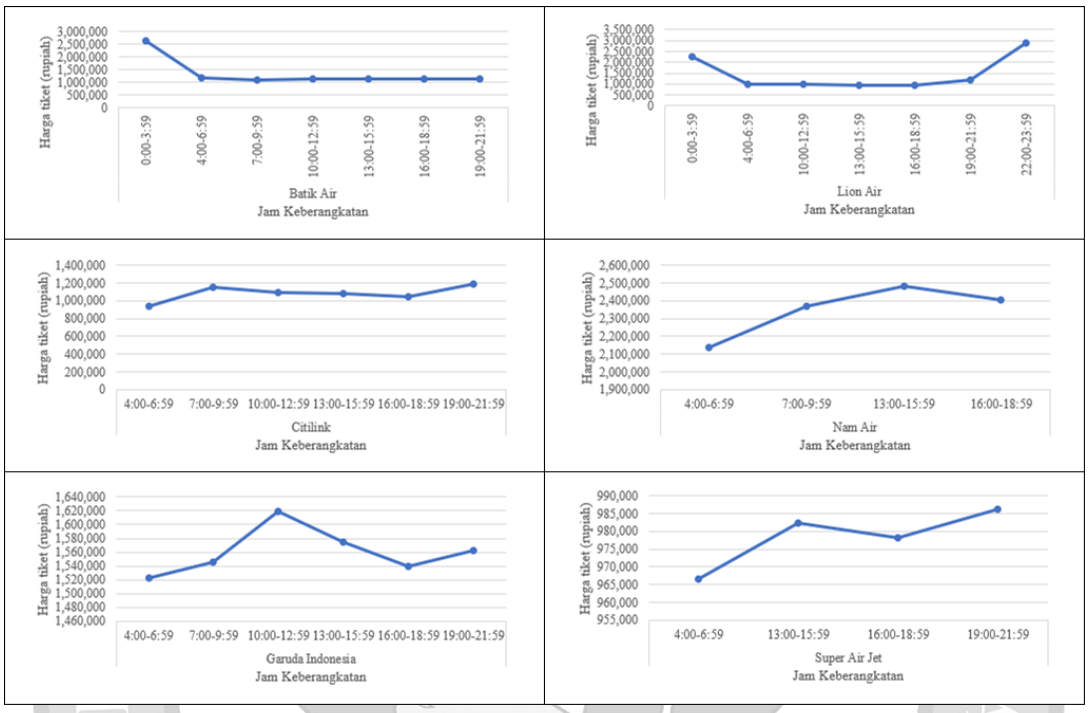
Gambar 4.3: Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Surabaya

4.2.1.2 Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan jam keberangkatan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.4. Pada maskapai penerbangan Batik Air, harga tiket pesawat paling mahal di jam keberangkatan 00.00-03.59 yaitu Rp2.632.500,00. Pada maskapai penerbangan Citilink, harga tiket pesawat paling mahal di jam keberangkatan 19.00-21.59, yaitu sebesar Rp1.191.729,00. Pada maskapai penerbangan Garuda Indonesia, harga tiket pesawat paling mahal di jam keberangkatan 10.00-12.59 yaitu Rp1.618.645,00.

Pada maskapai penerbangan Lion Air, harga tiket pesawat paling mahal ada di jam keberangkatan 22.00-23.59 yaitu Rp2.903.530,00. Pada maskapai penerbangan Nam Air, harga tiket pesawat paling mahal di jam keberangkatan 13.00-15.59 yaitu Rp2.481.526,00. Pada maskapai penerbangan Super Air Jet, harga tiket pesawat

paling mahal di jam 19.00-21.59 yaitu Rp986.280,00. Puncak harga tiket pesawat untuk setiap maskapai penerbangan ada di jam keberangkatan yang berbeda-beda sehingga tidak dapat disimpulkan jam keberangkatan dengan harga paling mahal untuk rute Jakarta-Surabaya.



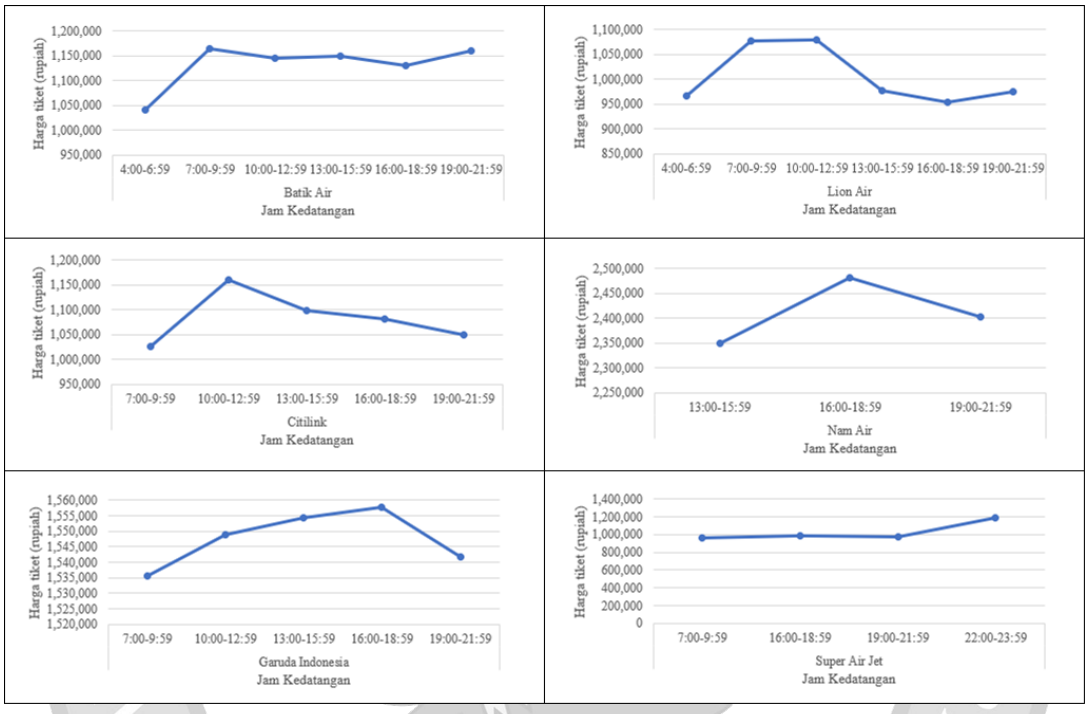
Gambar 4.4: Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Surabaya

4.2.1.3 Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan jam kedatangan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.5. Pada maskapai penerbangan Batik Air, harga tiket pesawat paling mahal di jam kedatangan 07.00-09.59 yaitu Rp1.164.158,00. Pada maskapai penerbangan Citilink, harga tiket pesawat paling mahal di jam kedatangan 10.00-12.59. Pada maskapai penerbangan Garuda Indonesia, harga tiket pesawat paling mahal di jam kedatangan 16.00-18.59 yaitu Rp1.557.721,00.

Pada maskapai penerbangan Lion Air, harga tiket pesawat paling mahal ada di jam kedatangan 10.00-12.59 yaitu Rp1.079.995,00, dan ini tidak jauh berbeda dari harga tiket pada jam keberangkatan 07.00-9.59 yaitu Rp1.078.128,00. Pada

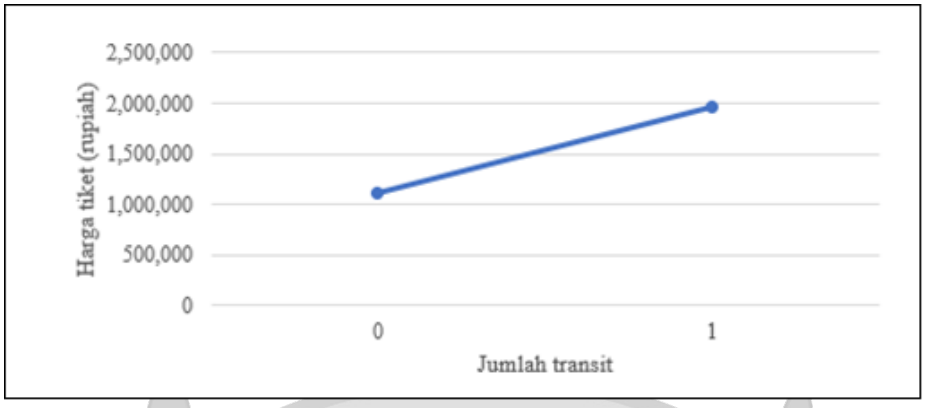
maskapai penerbangan Nam Air, harga tiket pesawat paling mahal di jam kedatangan 16.00-18.59 yaitu Rp2.481.526,00. Pada maskapai penerbangan Super Air Jet, harga tiket pesawat paling mahal di jam 22.00-23.59 yaitu Rp1.189.659,00. Puncak harga tiket pesawat untuk setiap maskapai penerbangan ada di jam kedatangan yang berbeda-beda sehingga tidak dapat disimpulkan jam kedatangan dengan harga paling mahal untuk rute Jakarta-Surabaya.



Gambar 4.5: Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Surabaya

4.2.1.4 Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan jumlah transit pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.6. Pada rute ini hanya terdapat penerbangan langsung dan penerbangan dengan satu transit. Penerbangan dengan satu transit memiliki rata-rata harga tiket sebesar Rp1.964.407,00, sedangkan penerbangan langsung memiliki rata-rata harga tiket sebesar Rp1.107.071,00. Terlihat harga tiket penerbangan dengan satu transit lebih mahal dari harga tiket penerbangan langsung dengan selisih hampir Rp900.000,00.



Gambar 4.6: Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Surabaya

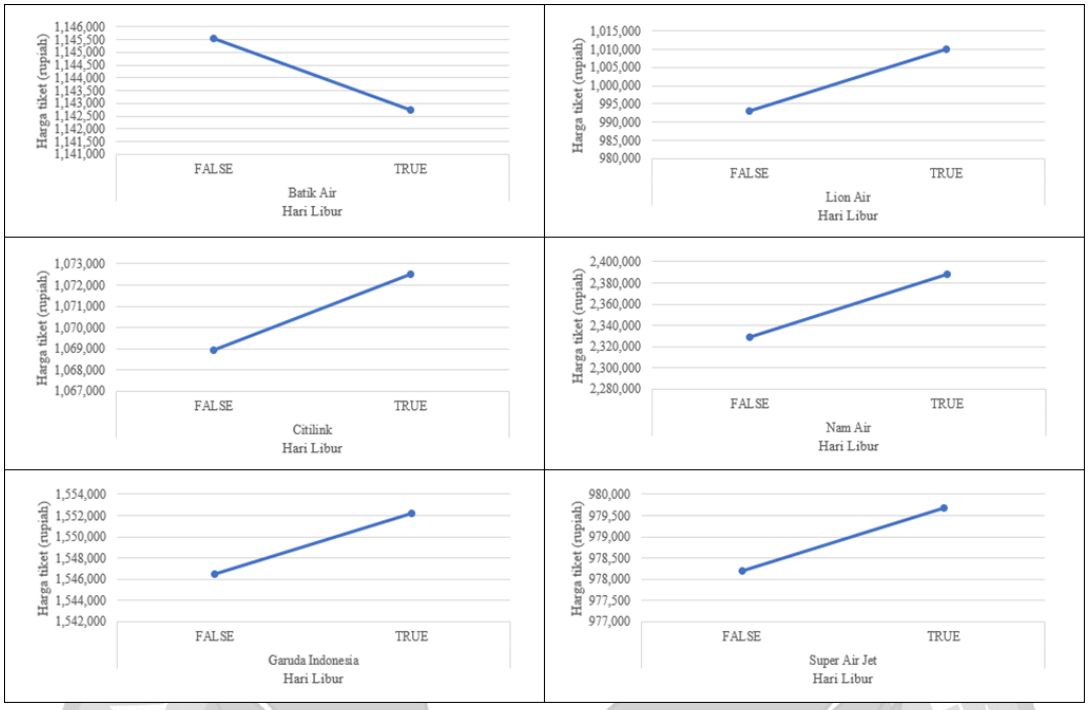
4.2.1.5 Harga Tiket Pesawat Berdasarkan Hari Penerbangan pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan hari penerbangan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.7. Pada maskapai penerbangan Batik Air, Citilink, Garuda Indonesia, Lion Air, dan Super Air Jet, rata-rata harga tiket pesawat tertinggi ada di hari Jumat. Hanya maskapai penerbangan Nam Air dengan rata-rata harga tiket tertinggi di hari Minggu yaitu Rp2.400.270,00, namun nilai ini tidak jauh berbeda dengan rata-rata harga tiket di hari Jumat yaitu sebesar Rp2.375.079,00. Dapat disimpulkan pada rute Jakarta-Surabaya, harga tiket pesawat cenderung lebih mahal di hari Jumat.

Gambar 4.7: Harga Tiket Pesawat Berdasarkan Hari Penerbangan pada Rute Jakarta-Surabaya

4.2.1.6 Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Surabaya

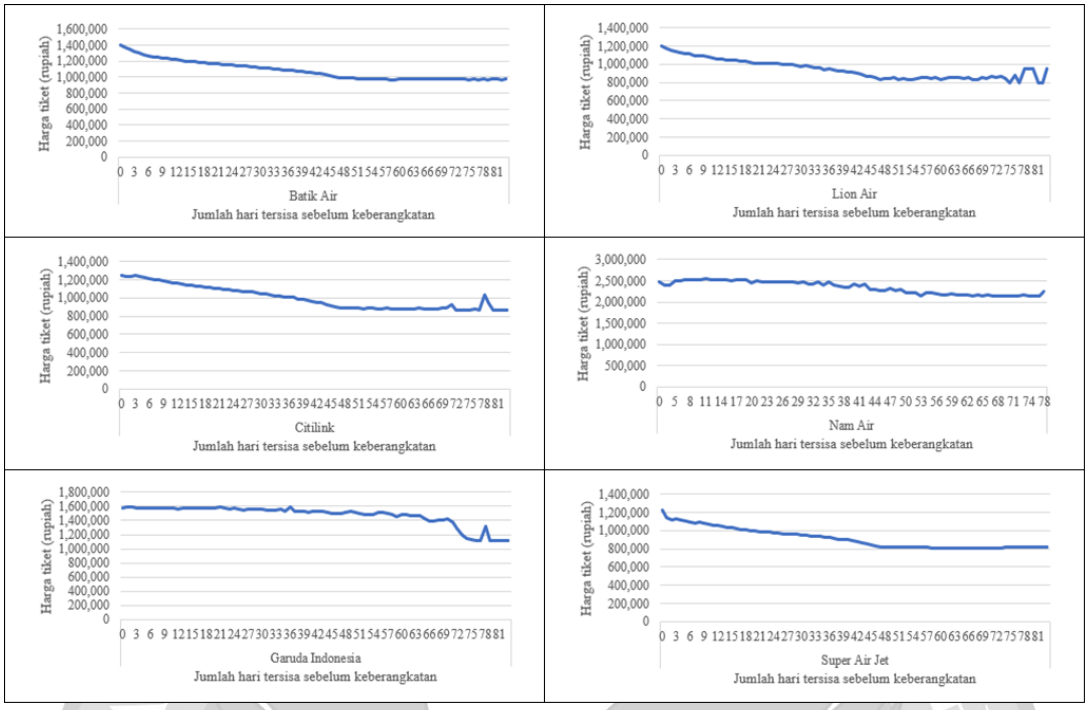
Harga tiket pesawat berdasarkan hari libur pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.8. Pada maskapai penerbangan Citilink, Garuda Indonesia, Lion Air, Nam Air, dan Super Air Jet, rata-rata harga tiket pesawat di hari libur lebih mahal daripada di hari bukan libur. Hanya maskapai penerbangan Batik Air dengan rata-rata harga tiket lebih tinggi di hari bukan libur yaitu Rp1.145.557,00, namun nilai ini tidak jauh berbeda dengan rata-rata harga tiket di hari Jumat yaitu sebesar Rp1.142.745,00. Dapat disimpulkan pada rute Jakarta-Surabaya, harga tiket pesawat cenderung lebih mahal di hari libur.



Gambar 4.8: Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Surabaya

4.2.1.7 Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Surabaya

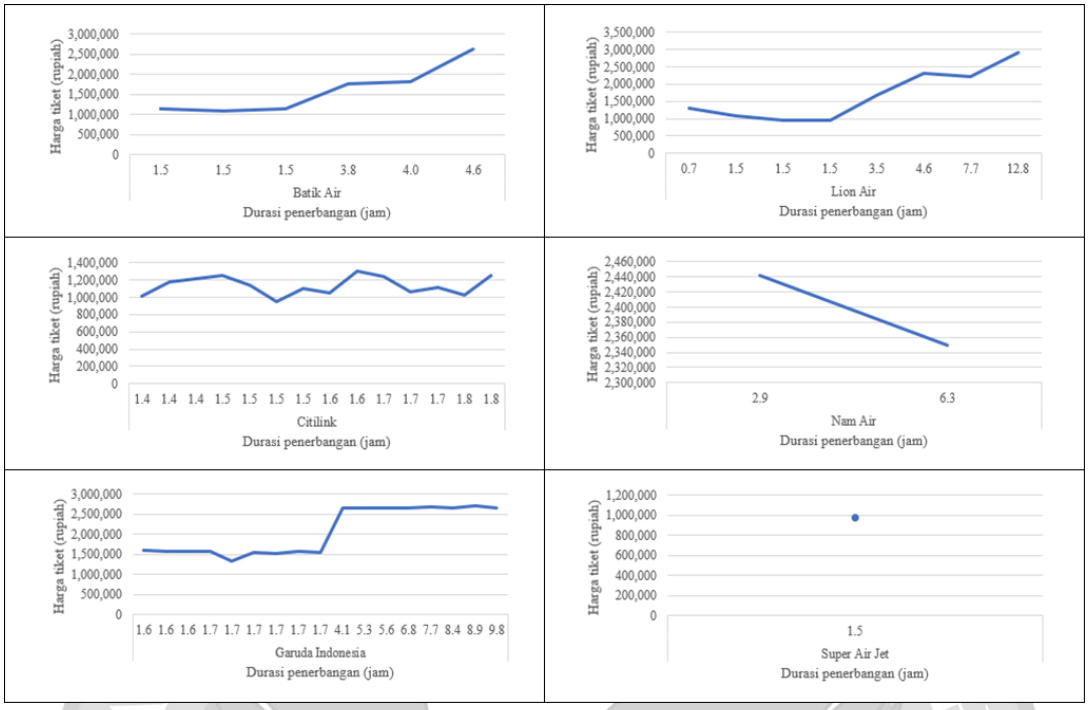
Harga tiket pesawat berdasarkan jumlah hari tersisa sebelum keberangkatan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.9. Terlihat bentuk garis pada setiap grafik menurun. Ini menunjukkan semakin besar jumlah hari tersisa sebelum keberangkatan, harga tiket pesawat cenderung semakin kecil. Dapat disimpulkan harga tiket pesawat pada seluruh maskapai penerbangan di rute Jakarta-Surabaya akan semakin naik jika berdekatan dengan hari keberangkatan.



Gambar 4.9: Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Surabaya

4.2.1.8 Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan durasi penerbangan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.10. Pada maskapai penerbangan Batik Air, Garuda Indonesia, dan Lion Air, rata-rata harga tiket pesawat akan semakin naik jika durasinya semakin lama. Berbeda dengan maskapai penerbangan Nam Air yang rata-rata harganya cenderung turun jika durasinya semakin lama. Pada maskapai penerbangan Citilink, harga tiket cenderung fluktuatif, sedangkan pada maskapai penerbangan Super Air Jet, hanya ada satu durasi penerbangan yaitu 1,5 jam.



Gambar 4.10: Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Surabaya

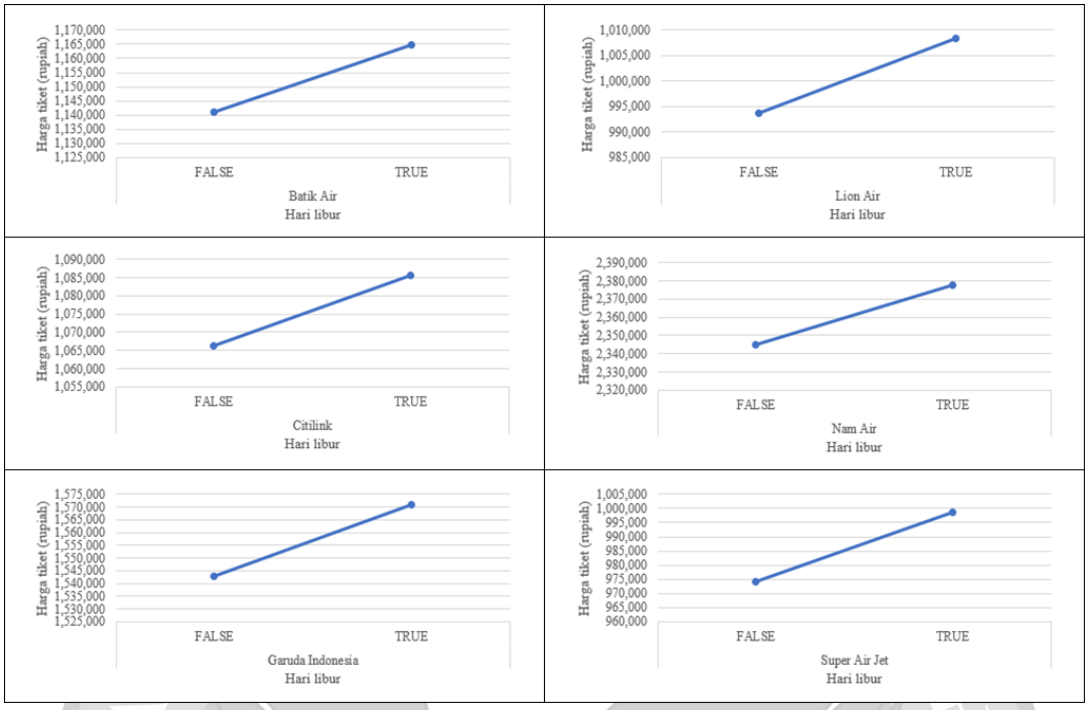
4.2.1.9 Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan hari data penerbangan didapatkan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.11. Hari data penerbangan didapatkan adalah hari di mana calon penumpang akan melakukan pembelian tiket pesawat. Pada seluruh maskapai penerbangan, rata-rata harga tiket pesawat berdasarkan hari data penerbangan didapatkan yang terendah ada di hari Kamis, kecuali maskapai penerbangan Nam Air. Hanya maskapai penerbangan Nam Air dengan rata-rata harga tiket terendah jika didapatkan di hari Selasa yaitu Rp2.294.352,00, namun nilai ini tidak jauh berbeda dengan rata-rata harga tiket yang didapatkan di hari Senin yaitu sebesar Rp2.317.789,00 dan di hari Kamis yaitu sebesar Rp2.323.057,00. Dapat disimpulkan pada rute Jakarta-Surabaya, harga tiket pesawat berdasarkan hari data penerbangan didapatkan cenderung lebih murah di hari Kamis.

Gambar 4.11: Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya

4.2.1.10 Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya

Harga tiket pesawat berdasarkan hari libur data penerbangan didapatkan pada rute Jakarta-Surabaya secara rata-rata ditunjukkan pada Gambar 4.12. Hari libur data penerbangan didapatkan ditentukan dari hari di mana calon penumpang akan melakukan pembelian tiket pesawat. Pada seluruh maskapai penerbangan, rata-rata harga tiket pesawat yang didapatkan di hari libur lebih mahal daripada di hari bukan libur. Dapat disimpulkan pada rute Jakarta-Surabaya, harga tiket pesawat cenderung lebih mahal jika didapatkan di hari libur.

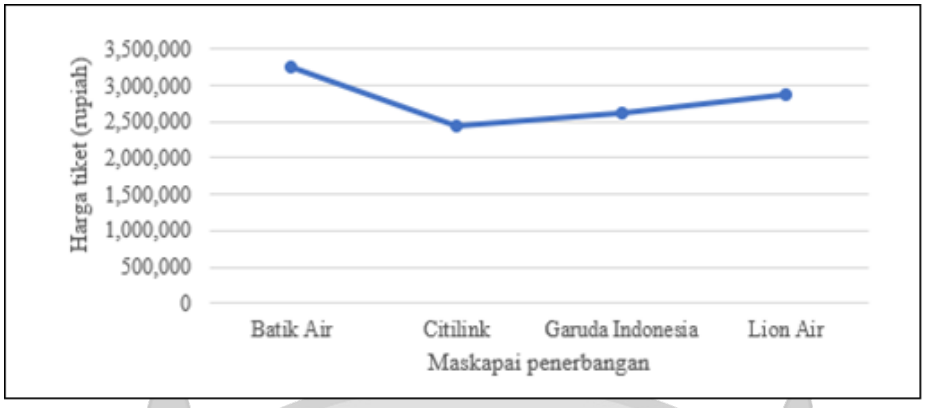


Gambar 4.12: Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Surabaya

4.2.2 Rute Jakarta-Manado (CGK-MDC)

4.2.2.1 Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Manado

Harga tiket pesawat berdasarkan jenis maskapai penerbangan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.13. Pada rute ini hanya ada empat maskapai penerbangan yaitu Batik Air, Citilink, Garuda Indonesia, dan Lion Air. Maskapai penerbangan dengan rata-rata harga tiket tertinggi adalah Batik Air, yaitu sebesar Rp3.262.777,00. Maskapai penerbangan dengan rata-rata harga tiket terendah adalah Citilink, yaitu sebesar Rp2.431.029,00.

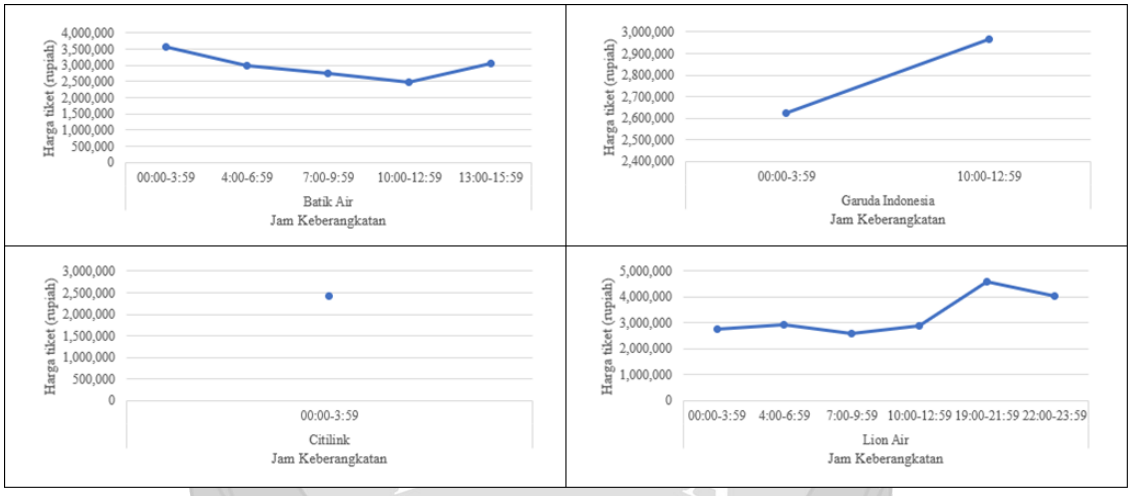


Gambar 4.13: Harga Tiket Pesawat Berdasarkan Jenis Maskapai Penerbangan pada Rute Jakarta-Manado

4.2.2.2 Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Manado

Harga tiket pesawat berdasarkan jam keberangkatan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.14. Pada maskapai penerbangan Batik Air, harga tiket pesawat paling mahal di jam keberangkatan 00.00-03.59 yaitu Rp3.582.417,00. Pada maskapai penerbangan Citilink, hanya ada satu jam keberangkatan yaitu pada jam 00.00-03.59 dengan rata-rata harga tiket sebesar Rp2.431.029,00.

Pada maskapai penerbangan Garuda Indonesia, harga tiket pesawat paling mahal di jam keberangkatan 10.00-12.59 yaitu Rp2.964.500,00. Pada maskapai penerbangan Lion Air, harga tiket pesawat paling mahal ada di jam keberangkatan 19.00-21.59 yaitu Rp4.585.488,00. Puncak harga tiket pesawat untuk setiap maskapai penerbangan ada di jam keberangkatan yang berbeda-beda sehingga tidak dapat disimpulkan jam keberangkatan dengan harga paling mahal untuk rute Jakarta-Manado.

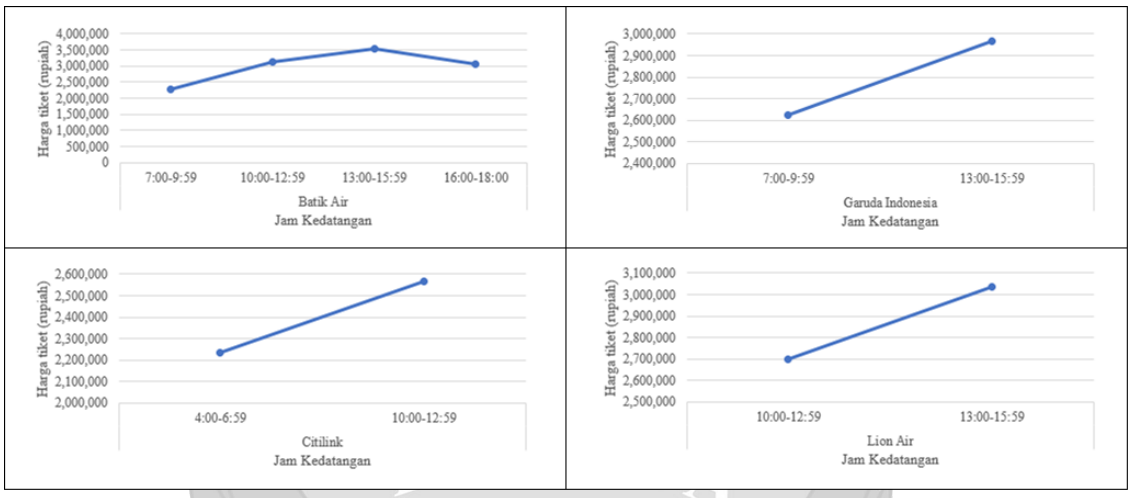


Gambar 4.14: Harga Tiket Pesawat Berdasarkan Jam Keberangkatan pada Rute Jakarta-Manado

4.2.2.3 Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Manado

Harga tiket pesawat berdasarkan jam kedatangan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.15. Pada maskapai penerbangan Batik Air, harga tiket pesawat paling mahal di jam kedatangan 13.00-15.59 yaitu Rp3.529.610,00. Pada maskapai penerbangan Citilink, harga tiket pesawat paling mahal di jam kedatangan 10.00-12.59, yaitu sebesar Rp2.565.283,00.

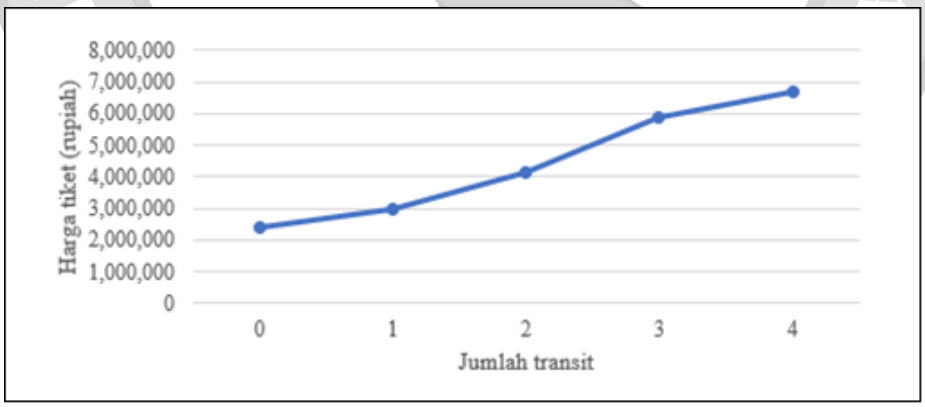
Pada maskapai penerbangan Garuda Indonesia, harga tiket pesawat paling mahal di jam kedatangan 13.00-15.59 yaitu Rp2.964.500,00. Pada maskapai penerbangan Lion Air, harga tiket pesawat paling mahal ada di jam kedatangan 13.00-15.59 yaitu Rp3.034.767,00. Puncak harga tiket pesawat untuk setiap maskapai penerbangan kecuali Citilink ada di jam kedatangan 13.00-15.59 sehingga dapat disimpulkan harga tiket pesawat cenderung mahal pada penerbangan dengan jam kedatangan 13.00-15.59.



Gambar 4.15: Harga Tiket Pesawat Berdasarkan Jam Kedatangan pada Rute Jakarta-Manado

4.2.2.4 Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Manado

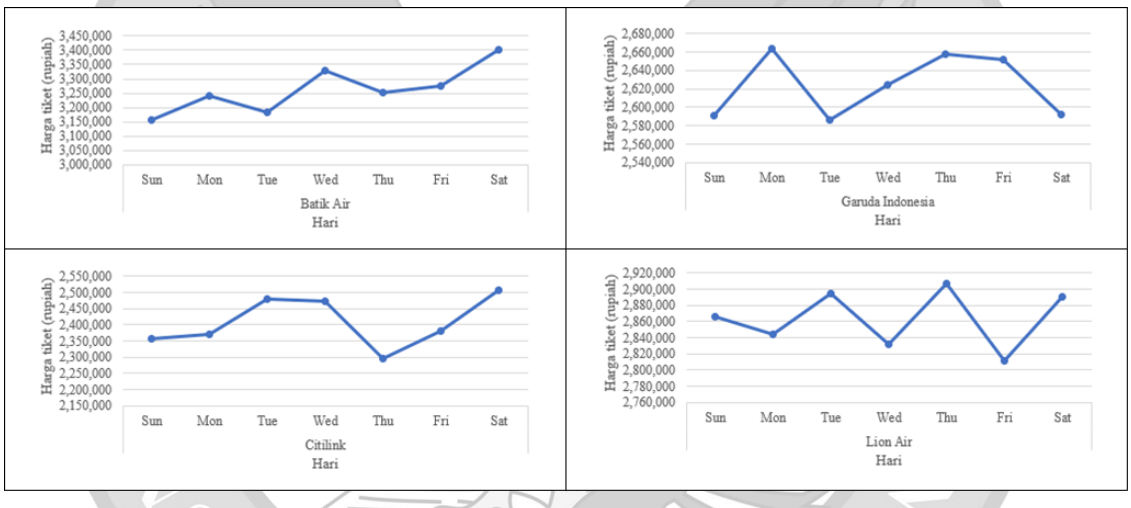
Harga tiket pesawat berdasarkan jumlah transit pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.16. Pada rute ini terdapat penerbangan langsung dan penerbangan dengan transit sebanyak satu kali hingga empat kali. Terlihat bentuk garis naik sehingga semakin tinggi jumlah transit, harga tiket pesawat pada rute ini akan semakin tinggi juga.



Gambar 4.16: Harga Tiket Pesawat Berdasarkan Jumlah Transit pada Rute Jakarta-Manado

4.2.2.5 Harga Tiket Pesawat Berdasarkan Hari Penerbangan pada Rute Jakarta-Manado

Harga tiket pesawat berdasarkan hari penerbangan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.17. Pada maskapai penerbangan Batik Air dan Citilink, rata-rata harga tiket pesawat tertinggi ada di hari Sabtu. Pada maskapai penerbangan Garuda Indonesia, rata-rata harga tiket pesawat tertinggi ada di hari Senin yaitu Rp2.663.265,00. Pada maskapai penerbangan Lion Air, rata-rata harga tertinggi ada di hari Kamis yaitu Rp2.907.277,00, dan nilai ini tidak jauh berbeda dengan rata-rata harga tiket di hari Selasa yaitu sebesar Rp2.894.234,00 dan di hari Sabtu yaitu sebesar Rp2.890.859,00. Dapat disimpulkan pada rute Jakarta-Manado, harga tiket pesawat cenderung lebih mahal di hari Sabtu.

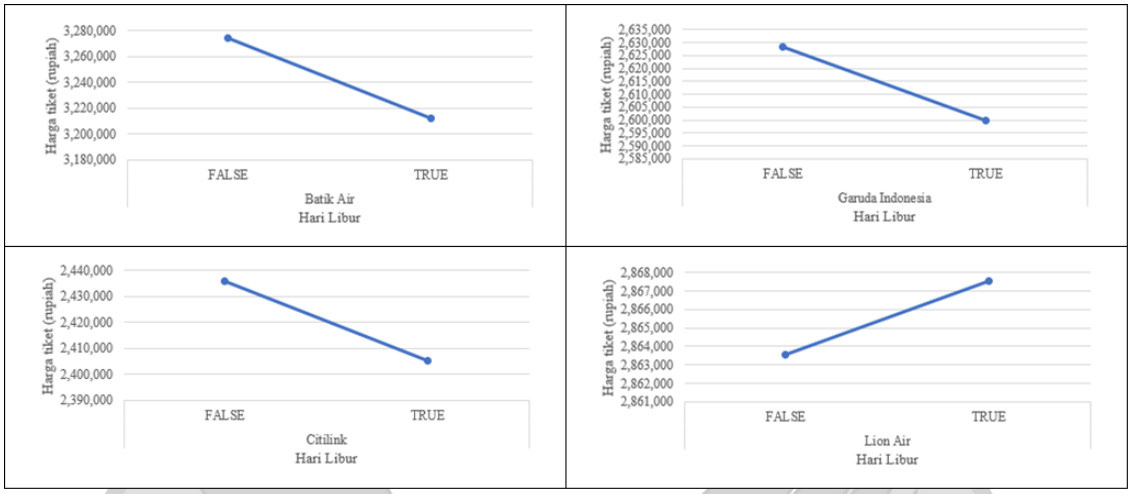


Gambar 4.17: Harga Tiket Pesawat Berdasarkan Hari Keberangkatan pada Rute Jakarta-Manado

4.2.2.6 Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Manado

Harga tiket pesawat berdasarkan hari libur pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.18. Pada maskapai penerbangan Batik Air, Citilink, dan Garuda Indonesia, rata-rata harga tiket pesawat di hari bukan libur lebih mahal daripada di hari libur. Hanya maskapai penerbangan Lion Air dengan rata-rata harga tiket lebih tinggi di hari libur yaitu Rp2.867.542,00, namun nilai ini tidak jauh berbeda dengan rata-rata harga tiket di hari Jumat yaitu sebesar

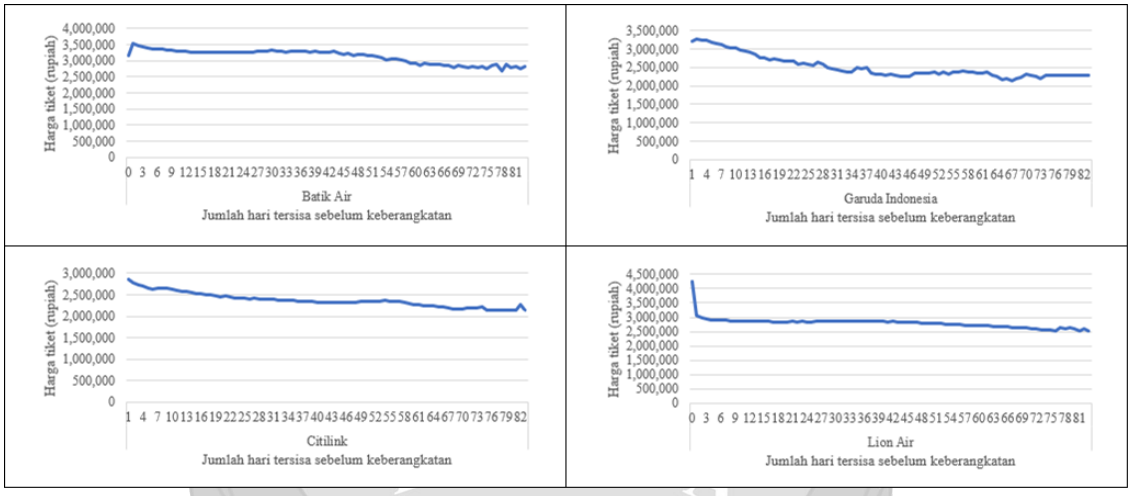
Rp2.863.561,00. Dapat disimpulkan pada rute Jakarta-Manado, harga tiket pesawat cenderung lebih mahal di hari bukan libur.



Gambar 4.18: Harga Tiket Pesawat Berdasarkan Hari Libur pada Rute Jakarta-Manado

4.2.2.7 Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Manado

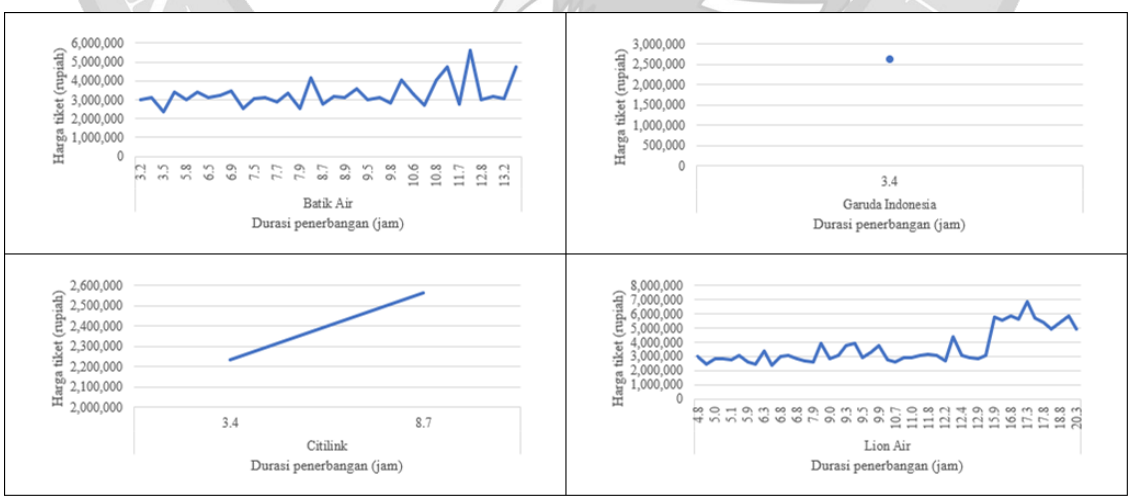
Harga tiket pesawat berdasarkan jumlah hari tersisa sebelum keberangkatan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.19. Terlihat bentuk garis pada setiap grafik menurun. Ini menunjukkan semakin besar jumlah hari tersisa sebelum keberangkatan, harga tiket pesawat cenderung semakin kecil. Dapat disimpulkan harga tiket pesawat pada seluruh maskapai penerbangan di rute Jakarta-Manado akan semakin naik jika berdekatan dengan hari keberangkatan.



Gambar 4.19: Harga Tiket Pesawat Berdasarkan Jumlah Hari Tersisa Sebelum Keberangkatan pada Rute Jakarta-Manado

4.2.2.8 Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Manado

Harga tiket pesawat berdasarkan durasi penerbangan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.20. Pada maskapai penerbangan Batik Air, Citilink, dan Lion Air, rata-rata harga tiket pesawat akan semakin naik jika durasinya semakin lama. Maskapai penerbangan Garuda Indonesia hanya memiliki satu durasi yaitu 3,4 jam. Dapat disimpulkan pada rute Jakarta-Manado, harga tiket pesawat cenderung naik jika durasinya semakin lama.



Gambar 4.20: Harga Tiket Pesawat Berdasarkan Durasi Penerbangan pada Rute Jakarta-Manado

4.2.2.9 Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Manado

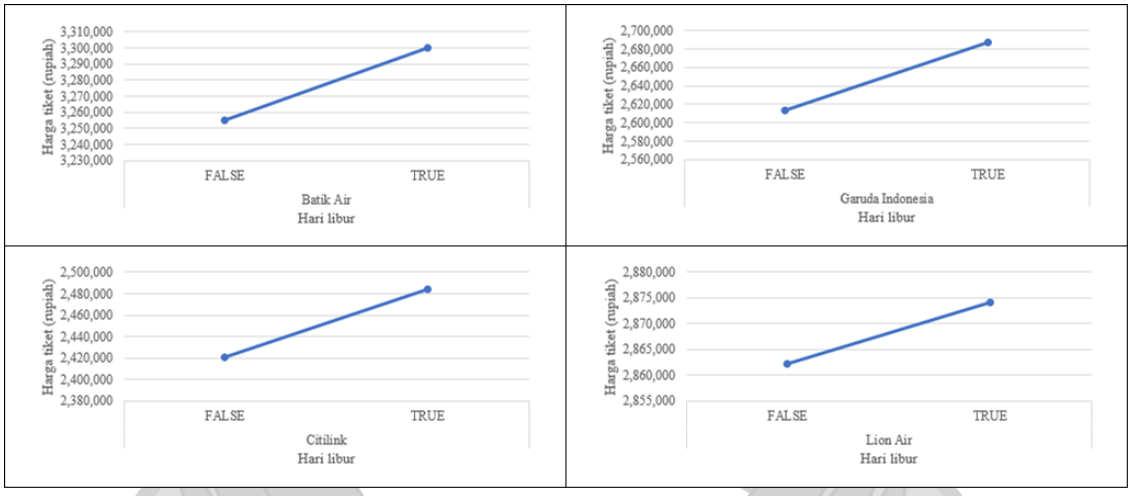
Harga tiket pesawat berdasarkan hari data penerbangan didapatkan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.21. Hari data penerbangan didapatkan adalah hari di mana calon penumpang akan melakukan pembelian tiket pesawat. Pada seluruh maskapai penerbangan, rata-rata harga tiket pesawat berdasarkan hari data penerbangan didapatkan yang terendah ada di hari Kamis, kecuali maskapai penerbangan Garuda Indonesia. Hanya maskapai penerbangan Garuda Indonesia dengan rata-rata harga tiket terendah jika didapatkan di hari Sabtu yaitu Rp2.561.155,00. Dapat disimpulkan pada rute Jakarta-Manado, harga tiket pesawat berdasarkan hari data penerbangan didapatkan cenderung lebih murah di hari Kamis.

Gambar 4.21: Harga Tiket Pesawat Berdasarkan Hari Data Penerbangan Didapatkan pada Rute Jakarta-Manado

4.2.2.10 Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Manado

Harga tiket pesawat berdasarkan hari libur data penerbangan didapatkan pada rute Jakarta-Manado secara rata-rata ditunjukkan pada Gambar 4.12. Hari libur data penerbangan didapatkan ditentukan dari hari di mana calon penumpang akan melakukan pembelian tiket pesawat. Pada seluruh maskapai penerbangan, rata-rata harga tiket pesawat yang didapatkan di hari libur lebih mahal daripada di hari bukan libur. Dapat disimpulkan pada rute Jakarta-Manado, harga tiket

pesawat cenderung lebih mahal jika didapatkan di hari libur.



Gambar 4.22: Harga Tiket Pesawat Berdasarkan Hari Libur Data Penerbangan Didapatkan pada Rute Jakarta-Manado

4.3 Pembagian Data

Model *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest* akan dibuat menggunakan data latihan kemudian diuji menggunakan data tes. Seluruh data yang telah dikumpulkan akan dibagi menjadi data latihan dan data tes dengan proporsi 80:20. Jumlah data latihan dan data tes untuk setiap rute adalah sebagai berikut.

1. Rute Jakarta-Surabaya: 81.117 data latihan dan 20.280 data tes.
2. Rute Jakarta-Manado: 43.597 data latihan dan 10.900 data tes.

4.4 Model Prediksi untuk Rute Jakarta-Surabaya

4.4.1 *Regression Tree*

4.4.1.1 Model *Regression Tree* Awal

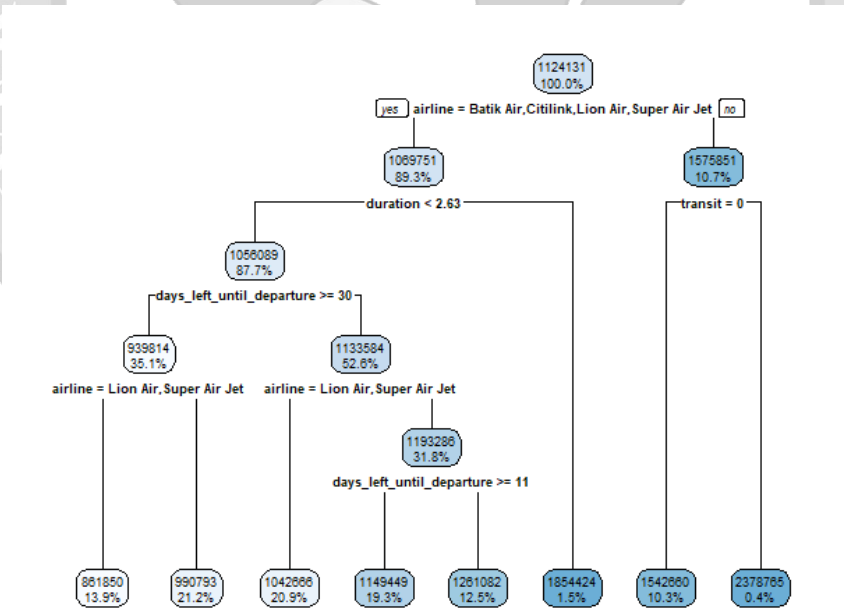
Pertama, dibuat model *Regression Tree* awal menggunakan fungsi `rpart` tanpa melakukan perubahan pada parameter `minsplit` dan `maxdepth`. Parameter `minsplit` adalah jumlah minimum nilai observasi yang harus ada di sebuah *node* agar *splitting* dapat dilakukan, sedangkan parameter `maxdepth` adalah maksimum

kedalaman dari sebuah model pohon. Nilai parameter yang digunakan merupakan nilai *default* sehingga nilai parameter `minsplit` adalah 20 dan nilai parameter `maxdepth` adalah 30. Setelah melakukan *fitting*, hasil model *Regression Tree* dapat dilihat pada *output* `m1`. Langkah-langkah *splitting* dari model `m1` adalah sebagai berikut.

```
> m1
n= 81117

node), split, n, deviance, yval
    * denotes terminal node

1) root 81117 6.199253e+15 1124131.0
    2) airline=Batik Air,Citilink,Lion Air,Super Air Jet 72401
       3.883359e+15 1069751.0
      4) duration< 2.625 71162 3.007692e+15 1056089.0
          8) days_left_until_departure>=29.5 28460 7.286184e+14
             939813.6
             16) airline=Lion Air,Super Air Jet 11252
                 2.219137e+14  861849.5 *
             17) airline=Batik Air,Citilink 17208 3.935890e+14
                 990792.9 *
          9) days_left_until_departure< 29.5 42702 1.637848e+15
             1133584.0
             18) airline=Lion Air,Super Air Jet 16926
                 5.197048e+14 1042666.0 *
             19) airline=Batik Air,Citilink 25776 8.863564e+14
                 1193286.0
                 38) days_left_until_departure>=10.5 15654
                     5.768392e+14 1149449.0 *
                 39) days_left_until_departure< 10.5 10122
                     2.329116e+14 1261082.0 *
```

```
    5) duration>=2.625 1239 9.951746e+13 1854424.0 *
  3) airline=Garuda Indonesia,NAM Air 8716 3.232861e+14
     1575851.0
    6) transit< 0.5 8370 7.877716e+13 1542660.0 *
    7) transit>=0.5 346 1.223179e+13 2378765.0 *
```

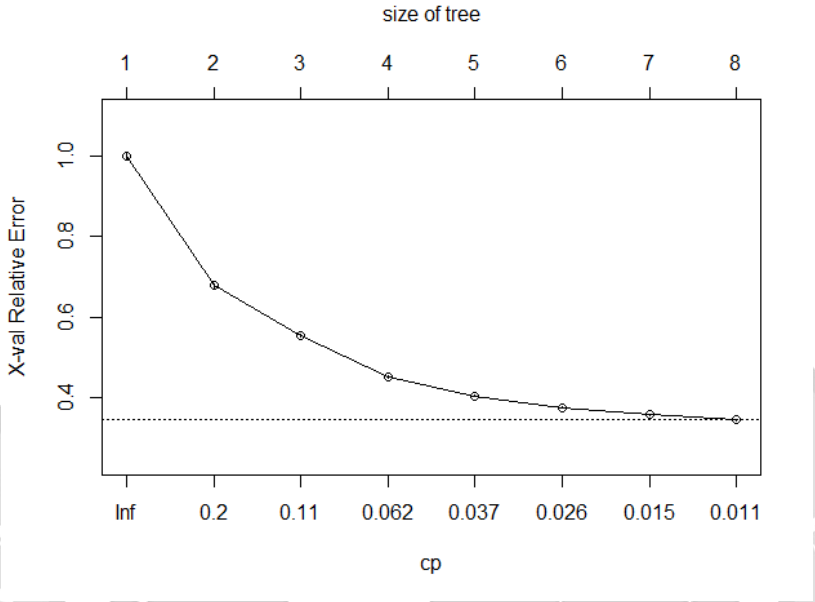
Model ini hanya menggunakan empat dari sepuluh variabel prediktor yaitu `airline`, `duration`, `transit`, dan `days_left_until_departure`. Model ini dimulai dengan 81.117 observasi pada *root node*. Variabel yang menjadi titik *splitting* pertama adalah `airline`, sehingga variabel paling penting yang memberikan pengurangan terbanyak di SSE pada awalnya adalah `airline`. Pada *node* pertama, sebanyak 72.401 observasi dengan `airline=Batik Air,Citilink,Lion Air,Super Air Jet` dibagi ke *node* kedua. Rata-rata harga tiket pesawat pada *node* ini adalah Rp1.069.751,00. Sisanya sebanyak 8.716 observasi dengan `airline=Garuda Indonesia,NAM Air` dibagi ke *node* ketiga. Rata-rata harga tiket pesawat pada *node* ini adalah Rp1.575.851,00. *Plot* model `m1` ditunjukkan pada Gambar 4.23.



Gambar 4.23: *Plot* Model *Regression Tree* Awal untuk Rute Jakarta-Surabaya

Model pohon yang ditunjukkan pada Gambar 4.23 terdapat 6 *internal nodes* dan 8 *terminal nodes*, sehingga $|T| = 8$. Fungsi `rpart` akan mempertimbangkan

seluruh nilai *complexity parameter* (CP) atau α untuk memangkas model pohon dan melakukan *10-fold* CV untuk setiap nilai α. Pada Gambar 4.24 ditunjukkan *plot* CP model *Regression Tree* awal.



Gambar 4.24: *Plot* CP Model *Regression Tree* Awal untuk Rute Jakarta-Surabaya

Dari Gambar 4.24, nilai setiap $|T|$ ditunjukkan pada sumbu `size of tree` dan setiap nilai α ditunjukkan pada sumbu `cp`, sedangkan setiap nilai eror dari *10-fold* CV ditunjukkan pada sumbu `X-val Relative Error`. Model pohon terbaik dengan nilai eror terendah ditunjukkan dengan garis putus-putus yaitu model dengan ukuran $|T| = 8$, $\alpha = 0,01$, dan nilai eror *10-fold* CV sebesar 0,34.

4.4.1.2 Model *Tuned Regression Tree*

Selanjutnya akan dilakukan *parameter tuning* pada parameter `minsplit` dan `maxdepth` dengan melakukan *grid search* untuk mendapatkan nilai parameter yang paling optimal. Nilai parameter `minsplit` terbaik akan dicari dari rentang 5 hingga 20, sedangkan nilai parameter `maxdepth` terbaik akan dicari dari rentang 5 hingga 15. Lima kombinasi parameter `minsplit` dan `maxdepth` terbaik adalah sebagai berikut.

```
> hyper_grid %>%
+   mutate(
+       cp      = purrr::map_dbl (models, get_cp),
+       error   = purrr::map_dbl (models, get_min_error)
+   ) %>%
+   arrange(error) %>%
+   top_n(-5, wt = error)
    minsplit   maxdepth   cp     error
1       20        14     0.01   0.3444384
2       10         8     0.01   0.3444396
3       16         5     0.01   0.3445284
4       10         7     0.01   0.3445442
5       20         5     0.01   0.3445569
```

Parameter `minsplit` dan `maxdepth` yang paling optimal adalah `minsplit=20` dan `maxdepth=14` dengan nilai $\alpha = 0,01$ dan nilai eror *10-fold* CV sebesar 0,34. Setelah didapatkan nilai parameter yang paling optimal, akan dibuat model baru dengan parameter `minsplit=20` dan `maxdepth=14`. Namun tidak ada perubahan dari model pertama, sehingga model akhir yang digunakan tetap model awal yang dijelaskan pada bagian 4.4.1.1.

Tingkat eror menggunakan MAPE pada setiap *terminal nodes* model *Regression Tree* ditunjukkan pada Tabel 4.2. Seluruh hasil prediksi pada model ini memiliki tingkat eror di bawah 20% sehingga berdasarkan Tabel 2.1, akurasi prediksi model ini baik. *Terminal node* ke-4 memiliki eror prediksi tertinggi yaitu sebesar 15,86%, sedangkan *terminal node* ke-7 dengan eror prediksi terendah yaitu sebesar 2,96%.

Tabel 4.2: Tingkat Kesalahan Prediksi pada Setiap *Terminal Nodes* Model *Regression Tree* untuk Rute Jakarta-Surabaya

Node	MAPE
1	10,86%
2	11,13%
3	15,54%
4	15,86%
5	9,75%
6	12,54%
7	2,96%
8	7,19%

4.4.1.3 Performa Model

Hasil *training error* dan *testing error* model *Regression Tree* untuk rute Jakarta-Surabaya ditunjukkan pada Tabel 4.3. Nilai *training error* dan *testing error* untuk model ini tidak saling jauh berbeda.

Tabel 4.3: Performa Model *Regression Tree* untuk Rute Jakarta-Surabaya

Performa	*Training Error*	*Testing Error*
MAE	129.715,8	129.764,2
MAPE	12,01%	12,04%
RMSE	162.252,8	162.198,0
R^2	65,55%	64,99%

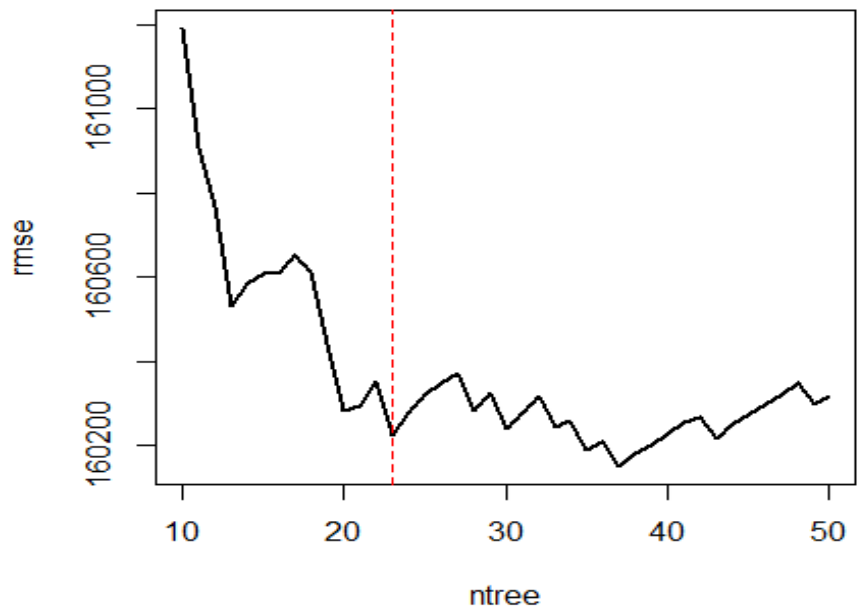
Berdasarkan nilai MAE dan MAPE, model ini cenderung memiliki kesalahan prediksi sebesar Rp129.764,20 atau 12,04% dari nilai sebenarnya secara rata-rata, sedangkan berdasarkan nilai RMSE, model ini cenderung memiliki kesalahan prediksi sebesar Rp162.198,00 secara rata-rata. Nilai R^2 menunjukkan bahwa 64,99% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.4.2 *Bagging Regression Tree*

4.4.2.1 Model *Bagging Regression Tree*

Pada *Bagging Regression Tree*, model akan dibuat menggunakan beberapa sampel *bootstrap*, sehingga akan dibuat beberapa model pohon. Parameter yang

mengatur jumlah model pohon adalah `nbagg`. Pada Gambar 4.25 ditunjukkan *plot* nilai eror prediksi terhadap jumlah model pohon.



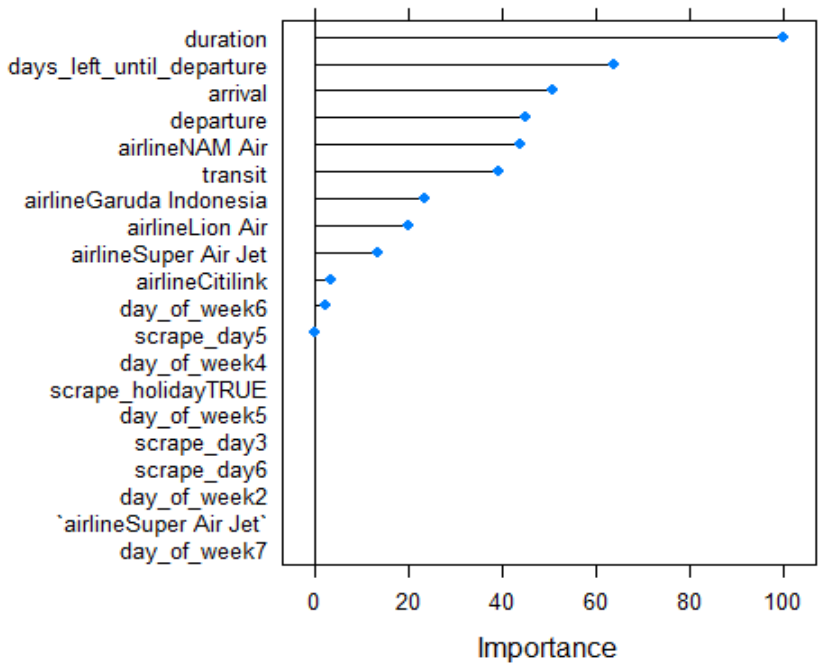
Gambar 4.25: *Plot* Jumlah Model Pohon terhadap Nilai Eror pada Model *Bagging Regression Tree* untuk Rute Jakarta-Surabaya

Nilai eror *out-of-bag* (OOB) akan dihitung menggunakan RMSE. Terlihat bahwa nilai eror mulai stabil saat jumlah model pohon sebanyak 23. Model tidak membaik secara signifikan dengan penambahan jumlah model pohon, sehingga model *Bagging Regression Tree* untuk rute Jakarta-Surabaya akan dibuat dengan hanya menggunakan 23 model pohon. Model *Bagging Regression Tree* untuk rute Jakarta-Surabaya beserta hasil *10-fold* CV untuk model ini adalah sebagai berikut.

```
> bagged_ml
Bagged CART

81117 samples
   10 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
```

```
Summary of sample sizes: 73006, 73006, 73005, 73004, 73005,
   73005, ...
Resampling results:


  RMSE      Rsquared  MAE
  161858.6  0.657228  131455.8
```

Berdasarkan nilai RMSE, model ini memiliki kesalahan prediksi sebesar Rp161.858,60 secara rata-rata, sedangkan berdasarkan nilai MAE, model ini memiliki kesalahan prediksi sebesar Rp131.455,80 secara rata-rata. Nilai R^2 menunjukkan bahwa 65,72% variabilitas pada variabel respons dapat dijelaskan oleh model ini. Tingkat *variable importance* ditunjukkan pada Gambar 4.26.



Gambar 4.26: Urutan *Variable Importance* pada Model *Bagging Regression Tree* untuk Rute Jakarta-Surabaya

Variabel `duration` adalah prediktor yang memberikan dampak terbesar terhadap SSE sehingga menjadi variabel terpenting. Selain variabel `duration`, variabel `days_left_until_departure` dan `arrival` juga memberikan dampak yang cukup besar pada SSE.

4.4.2.2 Performa Model

Hasil *training error* dan *testing error* model *Bagging Regression Tree* untuk rute Jakarta-Surabaya ditunjukkan pada Tabel 4.4. Nilai *training error* dan *testing error* untuk model ini tidak saling jauh berbeda.

Tabel 4.4: Performa Model *Bagging Regression Tree* untuk Rute Jakarta-Surabaya

Performa	*Training Error*	*Testing Error*
MAE	131.494,9	130.770,8
MAPE	12,42%	12,48%
RMSE	161.804,1	161.448,5
R^2	65,75%	65,31%

Berdasarkan nilai MAE dan MAPE, model ini cenderung memiliki kesalahan prediksi sebesar Rp130.626,70 atau 12,36% dari nilai sebenarnya secara rata-rata, sedangkan berdasarkan nilai RMSE, model ini cenderung memiliki kesalahan prediksi sebesar Rp160.892,80 secara rata-rata. Nilai R^2 menunjukkan bahwa 64,56% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.4.3 *Random Forest*

4.4.3.1 Model *Random Forest* Awal

Model *Random Forest* awal dibuat dengan menggunakan tiga parameter yaitu `mtry`, `ntree`, dan `maxnodes`. Parameter `mtry` adalah jumlah variabel yang akan diambil secara acak sebagai kandidat pada setiap *splitting*. Parameter `ntree` adalah jumlah model pohon yang akan dibuat, sedangkan parameter `maxnodes` adalah nilai maksimum *terminal nodes* pada setiap model pohon. Ditentukan nilai *default* untuk ketiga parameter pada model *Random Forest* yaitu `mtry=5`, `ntree=500`, dan `maxnodes=10`. Model *Random Forest* awal untuk rute Jakarta-Surabaya beserta hasil *10-fold* CV untuk model ini adalah sebagai berikut.

```
> print(rf_default)
Random Forest

81117 samples
   10 predictor

No pre-processing
Resampling: Cross-validated (5 fold)
Summary of sample sizes: 64894, 64894, 64894, 64894, 64892
Resampling results:

  RMSE      Rsquared    MAE
  182318.6  0.6238558   163549.6

Tuning parameter 'mtry' was held constant at a value of 5
```

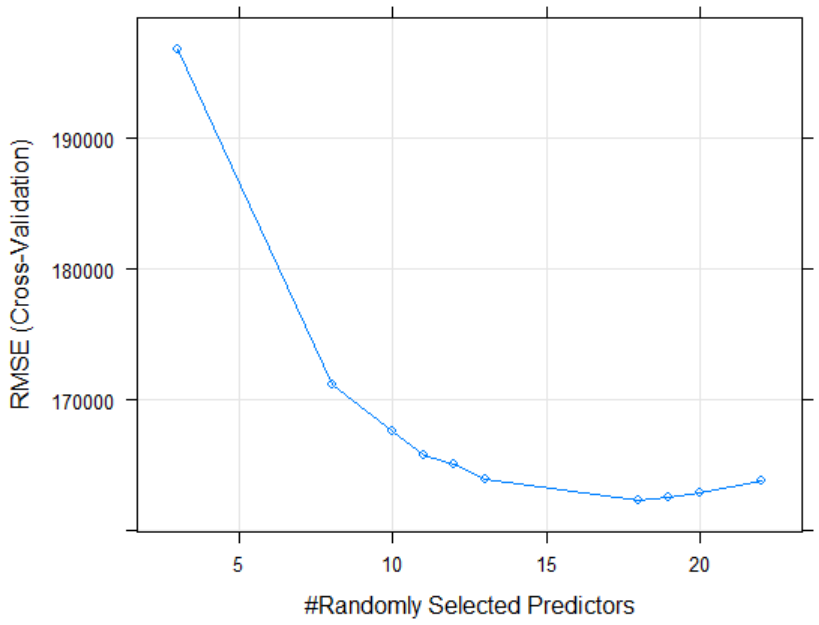
Berdasarkan nilai RMSE, model ini memiliki kesalahan prediksi sebesar Rp182.318,60 secara rata-rata, sedangkan berdasarkan nilai MAE, model ini memiliki kesalahan prediksi sebesar Rp163.549,60 secara rata-rata. Nilai R^2 menunjukkan bahwa 62,39% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.4.3.2 Model *Tuned Random Forest*

Parameter tuning akan dilakukan terhadap ketiga parameter yaitu `mtry`, `ntree`, dan `maxnodes`. Setiap nilai parameter akan dibandingkan nilai erornya menggunakan *5-fold* CV. Hasil *parameter tuning* terhadap `mtry` adalah sebagai berikut.

```
> print (rf_random)
Random Forest

81117 samples
   10 predictor

No pre-processing
Resampling: cross-validated (5 fold)

Sunmary of sanple sizes: 64894, 64894, 64894, 64893, 64893
Resampling results across tuning paraneters:

  mtry  RMSE        Rsquared    MAE
  3     196786.6    0.6189350   177468.8
  8     171194.0    0.6366929   149351.8
  10    167667.1    0.6444857   143804.0
  11    165787.6    0.6484243   140362.7
  12    165079.6    0.6503600   139246.4
  13    163963.3    0.6532927   136634.3
  18    162356.9    0.6566154   131403.6
  19    162618.0    0.6551012   131310.4
  20    162934.1    0.6534368   131755.3
  22    163820.2    0.6492257   132726.3

RMSE was used to select the optimal model using the smallest
    value. The final value used for the model was mtry = 18.
```

Parameter tuning terhadap `mtry` dilakukan menggunakan *random search*. Dipilih 10 nilai `mtry` secara acak dan untuk setiap nilai `mtry` akan dicari nilai erornya menggunakan *5-fold* CV. Terlihat pada Gambar 4.27 bahwa `mtry=18` adalah nilai yang paling optimal.



Gambar 4.27: *Plot* Hasil *Parameter Tuning* terhadap Parameter *mtry* pada Model *Random Forest* untuk Rute Jakarta-Surabaya

Parameter tuning terhadap `ntree` dilakukan secara manual menggunakan *looping*. Terdapat tiga nilai `ntree` yang akan dibandingkan yaitu 200, 500, dan 1000. Dari ketiga nilai tersebut, akan dicari yang paling optimal. Pada Tabel 4.5 ditampilkan hasil *parameter tuning* terhadap `ntree`. Terlihat bahwa `ntree=1000` adalah nilai yang paling optimal.

Tabel 4.5: *Parameter Tuning* terhadap Parameter *ntree* pada Model *Random Forest* untuk Rute Jakarta-Surabaya

	MAE	RMSE	*R-Squared*
200	131.638,9	162.572,3	0,6555812
500	131.495,9	162.498,3	0,6558983
1000	131.424,9	162.460,4	0,6560491

Parameter tuning terhadap `maxnodes` dilakukan secara manual menggunakan *looping*. Terdapat tiga nilai `maxnodes` yang akan dibandingkan yaitu 5, 10, 20, 30, 40, 50. Dari keenam nilai tersebut, akan dicari yang paling optimal. Pada Tabel 4.6 ditampilkan hasil *parameter tuning* terhadap `maxnodes`. Terlihat bahwa `maxnodes=50` adalah nilai yang paling optimal.

Tabel 4.6: *Parameter Tuning* terhadap Parameter *maxnodes* pada Model *Random Forest* untuk Rute Jakarta-Surabaya

	MAE	RMSE	*R-Squared*
5	144.994,7	176.191,8	0,5985680
10	131.424,9	162.460,4	0,6560491
20	124.957,8	155.043,8	0,6861278
30	122.933,4	152.868,9	0,6947357
40	120.160,9	150.718,9	0,7029371
50	119.815,0	150.341,4	0,7044091

Dari hasil *parameter tuning*, didapatkan nilai yang paling optimal untuk model *Random Forest* yaitu `mtry=18`, `ntree=1000`, dan `maxnodes=50`. Model *Tuned Random Forest* untuk rute Jakarta-Surabaya adalah sebagai berikut.

```
> rf_opt
Random Forest

81117 samples
   10 predictor

No pre-processing
Resampling: Cross-validated (5 fold)
Summary of sample sizes: 64894, 64892, 64894, 64894, 64892
Resampling results:

  RMSE      Rsquared    MAE
  150309    0.7043673   119775.9

Tuning parameter 'mtry' was held constant at a value of 18
```

Berdasarkan nilai RMSE, model ini memiliki kesalahan prediksi sebesar Rp150.309,00 secara rata-rata, sedangkan berdasarkan nilai MAE, model ini memiliki kesalahan prediksi sebesar Rp119.775,90 secara rata-rata. Nilai R^2 menunjukkan bahwa 70,44% variabilitas pada variabel respons dapat dijelaskan oleh model ini. Ketiga nilai tersebut jauh lebih baik daripada model *Random Forest* awal yang dijelaskan pada bagian 4.4.3.1, sehingga model akhir yang digunakan adalah model *Tuned Random Forest*.

4.4.3.3 Performa Model

Hasil *training error* dan *testing error* model *Random Forest* untuk rute Jakarta-Surabaya ditunjukkan pada Tabel 4.7. Nilai *training error* dan *testing error* untuk model ini tidak saling jauh berbeda.

Tabel 4.7: Performa Model *Random Forest* untuk Rute Jakarta-Surabaya

Performa	*Training Error*	*Testing Error*
MAE	119.604,5	120.257,0
MAPE	11,16%	11,23%
RMSE	150.037,0	150.536,2
R^2	70,57%	69,87%

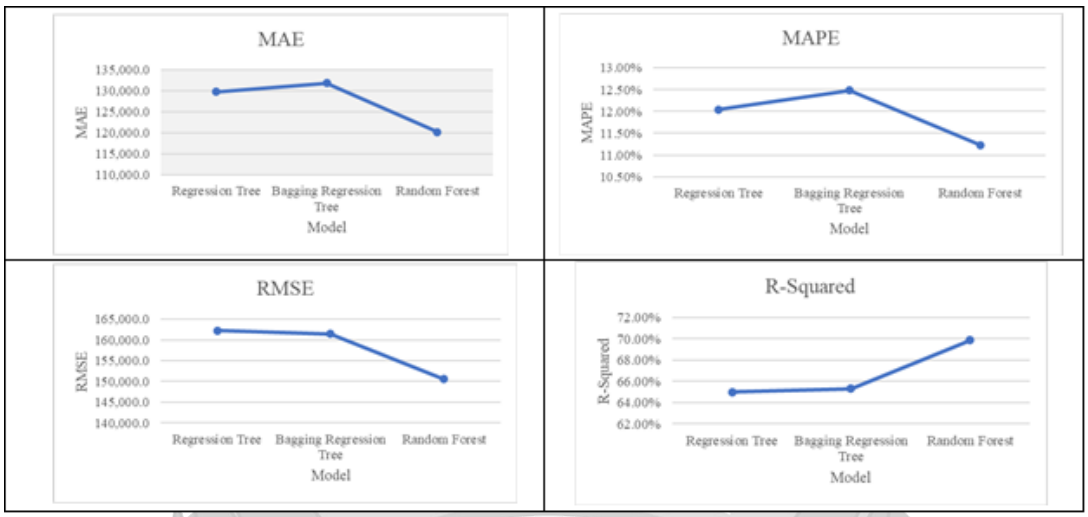
Berdasarkan nilai MAE dan MAPE, model ini cenderung memiliki kesalahan prediksi sebesar Rp120.257,00 atau 11,23% dari nilai sebenarnya secara rata-rata, sedangkan berdasarkan nilai RMSE, model ini cenderung memiliki kesalahan prediksi sebesar Rp150.536,20 secara rata-rata. Nilai R^2 menunjukkan bahwa 69,87% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.4.4 Evaluasi Performa Seluruh Model

Performa tiap model akan dibandingkan melalui kemampuan tiap model dalam melakukan prediksi. Nilai *testing error* setiap model ditunjukkan pada tabel 4.8 dan divisualisasikan pada Gambar 4.28.

Tabel 4.8: Performa Model untuk Rute Jakarta-Surabaya

Model	MAE	MAPE	RMSE	R^2
Regression Tree	129.764,2	12,04%	162.198,0	64,99%
Bagging Regression Tree	130.626,7	12,36%	160.892,8	65,56%
Random Forest	120.257,0	11,23%	150.536,2	69,87%



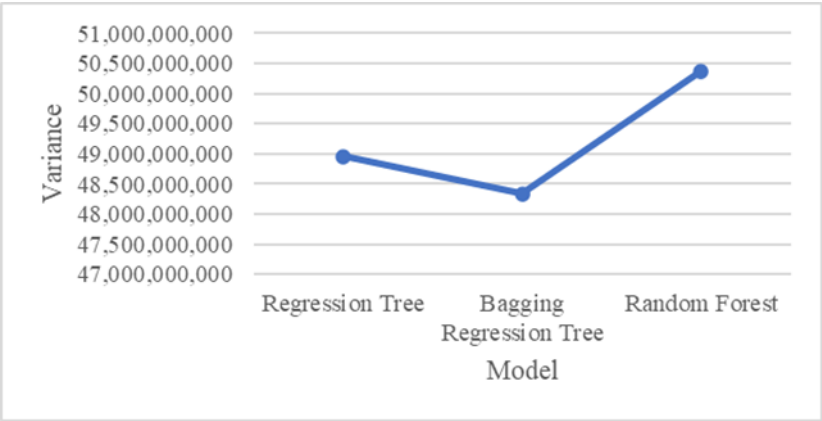
Gambar 4.28: Visualisasi Performa Model untuk Rute Jakarta-Surabaya

Model *Random Forest* memiliki nilai MAE, MAPE, dan RMSE terendah. Model *Random Forest* juga memiliki nilai R^2 tertinggi, sehingga model terbaik untuk rute Jakarta-Surabaya adalah model *Random Forest*. Model *Regression Tree* dan *Bagging Regression Tree* tidak berbeda secara signifikan, namun model *Regression Tree* memiliki nilai MAE dan MAPE yang lebih baik, sedangkan model *Bagging Regression Tree* memiliki nilai RMSE dan R^2 yang lebih baik. RMSE memberikan penalti yang lebih besar pada nilai eror yang besar, sehingga model *Regression Tree* cenderung menghasilkan nilai eror yang besar dibandingkan dengan model *Bagging Regression Tree*.

Nilai bias dan varians ditunjukkan pada Tabel 4.9. Nilai varians pada setiap model jauh lebih besar dibandingkan nilai biasnya karena ketiga model tersebut merupakan model nonparametrik yang fleksibel. Model *Bagging Regression Tree* dan *Random Forest* dilakukan untuk mengurangi varians dibandingkan model *Regression Tree*. Namun berdasarkan hasil yang didapatkan, hanya model *Bagging Regression Tree* yang dapat mengurangi varians, sedangkan model *Random Forest* memiliki varians yang tinggi seperti yang ditunjukkan pada Gambar 4.29

Tabel 4.9: Nilai Bias dan Varians Model Prediksi untuk Rute Jakarta-Surabaya

Model	Bias	Varians
Regression Tree	309.791,8	48.952.373.793
Bagging Regression Tree	189.953,1	48.341.483.612
Random Forest	340.962,5	50.371.954.989



Gambar 4.29: Nilai Varians Model Prediksi untuk Rute Jakarta-Surabaya

4.5 Model Prediksi untuk Rute Jakarta-Manado

4.5.1 *Regression Tree*

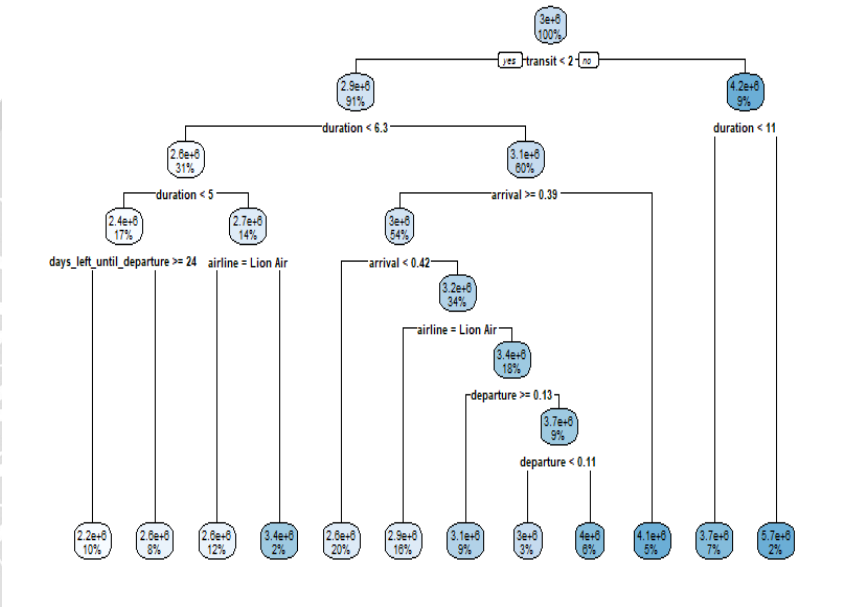
4.5.1.1 Model *Regression Tree* Awal

Pertama, dibuat model *Regression Tree* awal menggunakan fungsi `rpart` tanpa melakukan perubahan pada parameter. Nilai parameter yang digunakan adalah yang *default* sehingga nilai parameter `minsplit` adalah 20 dan nilai parameter `maxdepth` adalah 30. Setelah melakukan *fitting*, hasil model *Regression Tree* dapat dilihat pada *output* `m1`. Langkah-langkah *splitting* dari model `m1` adalah sebagai berikut.

```
> m1
n= 43597

node), split, n, deviance, yval
      * denotes terminal node

  1) root 43597 2.382562e+16 3011970
    2) transit< 1.5 39663 1.428367e+16 2893626
      4) duration< 6.25 13711 2.069518e+15 2552969
        8) duration< 4.958333 7555 9.624809e+14 2403579
          16) days_left_until_departure>=23.5 4272 2.467771e+14
              2221711 *
          17) days_left_until_departure< 23.5 3283 3.905366e+14
              2640234 *

        9) duration>=4.958333 6156 7.315013e+14 2736310
          18) airline=Lion Air 5309 2.926277e+14 2631251 *
          19) airline=Batik Air 847 1.298258e+13 3394824 *
      5) duration>=6.25 25952 9.782414e+15 3073602
        10) arrival>=0.39 23756 6.380082e+15 2975216
          20) arrival< 0.42 8761 5.463487e+14 2643918 *
          21) arrival>=0.42 14995 4.310317e+15 3168781
            42) airline=Lion Air 6965 4.511212e+14 2876006 *
            43) airline=Batik Air 8030 2.744337e+15 3422726
              86) departure>=0.13 3932 1.910033e+14 3126630 *
              87) departure< 0.13 4098 1.877838e+15 3706828
                174) departure< 0.105 1359 5.763900e+13 3017992 *
                175) departure>=0.105 2739 8.554121e+14 4048605 *
        11) arrival< 0.39 2196 6.847918e+14 4137924 *
    3) transit>=1.5 3934 3.385890e+15 4205130
      6) duration< 11 2966 3.712277e+14 3731573 *
      7) duration>=11 968 3.114796e+14 5656133 *
```

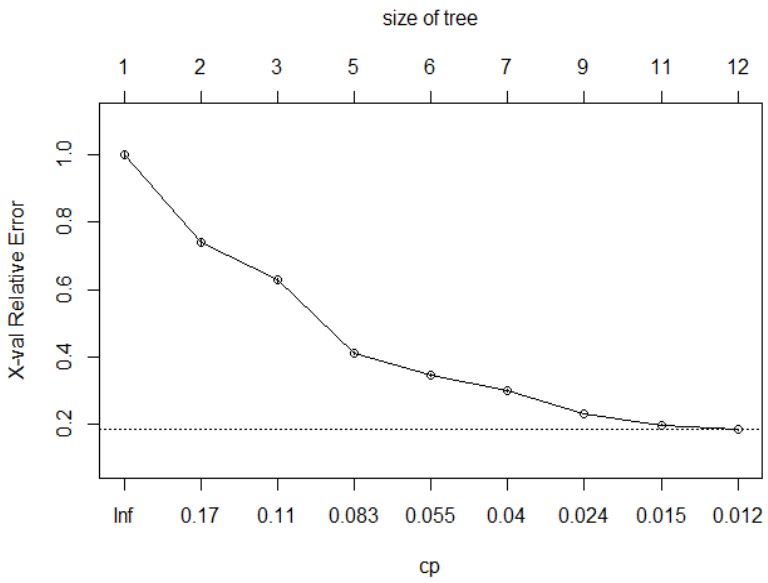
Model ini hanya menggunakan enam dari sepuluh variabel prediktor yaitu

`transit`, `duration`, `arrival`, `days_left_until_departure`, `airline`, dan `departure`. Model ini dimulai dengan 43.597 observasi pada *root node*. Variabel yang menjadi titik *splitting* pertama adalah `transit`, sehingga variabel paling penting yang memberikan pengurangan terbanyak di SSE pada awalnya adalah `transit`. Pada *node* pertama, sebanyak 39.663 observasi dengan `transit<1.5` dibagi ke *node* kedua. Rata-rata harga tiket pesawat pada *node* ini adalah Rp2.893.626,00. Sisanya sebanyak 3934 observasi dengan `transit>=1.5` dibagi ke *node* ketiga. Rata-rata harga tiket pesawat pada *node* ini adalah Rp4.205.130,00. *Plot* model `m1` ditunjukkan pada Gambar 4.30.



Gambar 4.30: *Plot* Model *Regression Tree* Awal untuk Rute Jakarta-Manado

Model pohon yang ditunjukkan pada Gambar 4.30 terdapat 10 *internal nodes* dan 12 *terminal nodes*, sehingga $|T| = 12$. Fungsi `rpart` akan mempertimbangkan seluruh nilai *complexity parameter* (CP) atau α untuk memangkas model pohon dan melakukan *10-fold* CV untuk setiap nilai α. Pada Gambar 4.31 ditunjukkan *plot* CP model *Regression Tree* awal.



Gambar 4.31: *Plot* CP Model *Regression Tree* Awal untuk Rute Jakarta-Manado

Dari Gambar 4.31, nilai setiap $|T|$ ditunjukkan pada sumbu `size of tree` dan setiap nilai α ditunjukkan pada sumbu `cp`, sedangkan setiap nilai eror dari *10-fold* CV ditunjukkan pada sumbu `X-val Relative Error`. Model pohon terbaik dengan nilai eror terendah ditunjukkan dengan garis putus-putus yaitu model dengan ukuran $|T| = 12$, $\alpha = 0,01$, dan nilai eror *10-fold* CV sebesar 0,19.

4.5.1.2 Model *Tuned Regression Tree*

Selanjutnya akan dilakukan *parameter tuning* pada parameter `minsplit` dan `maxdepth` dengan melakukan *grid search* untuk mendapatkan nilai parameter yang paling optimal. Nilai parameter `minsplit` terbaik akan dicari dari rentang 5 hingga 20, sedangkan nilai parameter `maxdepth` terbaik akan dicari dari rentang 5 hingga 15. Lima kombinasi parameter `minsplit` dan `maxdepth` terbaik adalah sebagai berikut.

```
> hyper_grid %>%
+   mutate(
+       cp      = purrr::map_dbl (models, get_cp),
+       error   = purrr::map_dbl (models, get_min_error)
+   ) %>%
+   arrange(error) %>%
+   top_n(-5, wt = error)
    minsplit    maxdepth    cp      error
1       17          13      0.01    0.1853437
2       7           14      0.01    0.1853635
3       5           15      0.01    0.1853840
4       14          9       0.01    0.1853998
5       12          14      0.01    0.1854109
```

Parameter `minsplit` dan `maxdepth` yang paling optimal adalah `minsplit=17` dan `maxdepth=13` dengan nilai $\alpha = 0,01$ dan nilai eror *10-fold* CV sebesar 0,34. Setelah didapatkan nilai parameter yang paling optimal, akan dibuat model baru dengan parameter `minsplit=17` dan `maxdepth=13`. Namun tidak ada perubahan dari model pertama, sehingga model akhir yang digunakan tetap model awal yang dijelaskan pada bagian 4.5.1.1.

Tingkat eror menggunakan MAPE pada setiap *terminal nodes* model *Regression Tree* ditunjukkan pada Tabel 4.10. Seluruh hasil prediksi pada model ini memiliki tingkat eror di bawah 15% sehingga berdasarkan Tabel 2.1, akurasi prediksi model ini baik. *Terminal node* ke-9 memiliki eror prediksi tertinggi yaitu sebesar 12,13%, sedangkan *terminal node* ke-4 dengan eror prediksi terendah yaitu sebesar 3,01%.

Tabel 4.10: Tingkat Kesalahan Prediksi pada Setiap *Terminal Nodes* Model *Regression Tree* untuk Rute Jakarta-Manado

Node	MAPE
1	8,61%
2	11,21%
3	7,28%
4	3,01%
5	7,65%
6	6,65%
7	5,47%
8	5,47%
9	12,13%
10	12,01%
11	6,76%
12	8,86%

4.5.1.3 Performa Model

Hasil *training error* dan *testing error* model *Regression Tree* untuk rute Jakarta-Manado ditunjukkan pada Tabel 4.11. Nilai *training error* dan *testing error* untuk model ini tidak saling jauh berbeda.

Tabel 4.11: Performa Model *Regression Tree* untuk Rute Jakarta-Manado

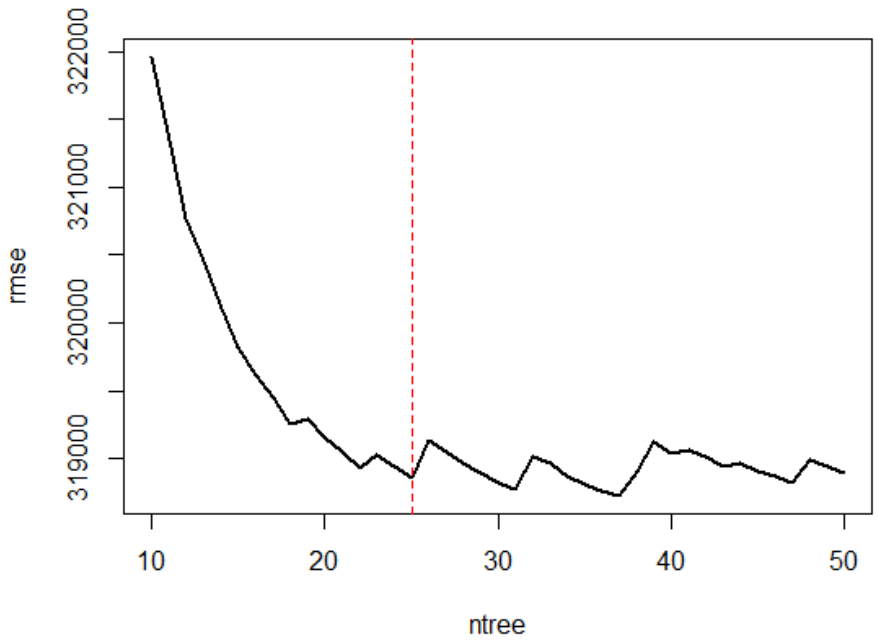
Performa	*Training Error*	*Testing Error*
MAE	244.682,5	248.163,6
MAPE	8,06%	8,14%
RMSE	318.117,0	322.838,7
R^2	81,48%	81,01%

Berdasarkan nilai MAE dan MAPE, model ini cenderung memiliki kesalahan prediksi sebesar Rp248.163,60 atau 8,14% dari nilai sebenarnya secara rata-rata, sedangkan berdasarkan nilai RMSE, model ini cenderung memiliki kesalahan prediksi sebesar Rp322.838,70 secara rata-rata. Nilai R^2 menunjukkan bahwa 81,01% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.5.2 *Bagging Regression Tree*

4.5.2.1 Model *Bagging Regression Tree*

Pada *Bagging Regression Tree*, model akan dibuat menggunakan beberapa sampel *bootstrap*, sehingga akan dibuat beberapa model pohon. Parameter yang mengatur jumlah model pohon adalah `nbagg`. Pada Gambar 4.32 ditunjukkan *plot* nilai eror prediksi terhadap jumlah model pohon.

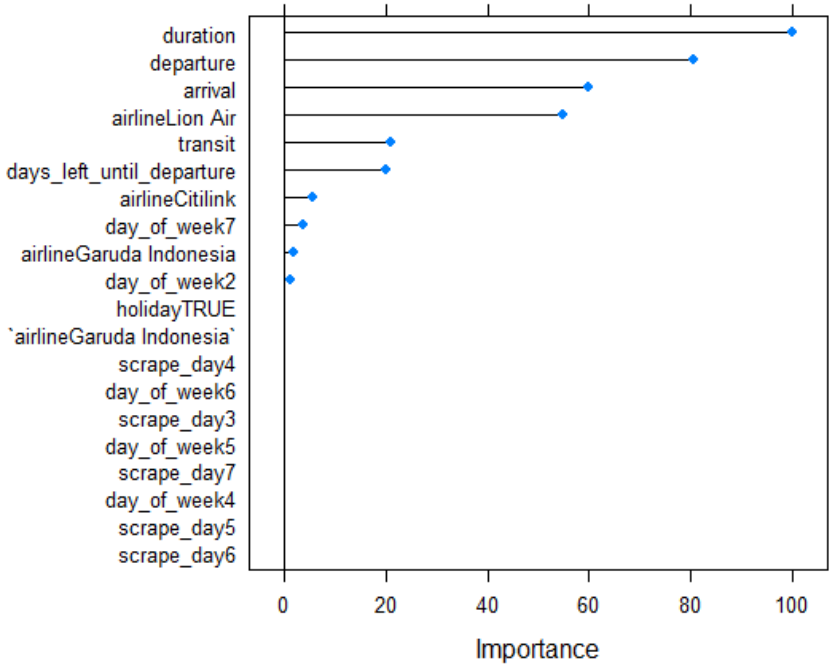


Gambar 4.32: *Plot* Jumlah Model Pohon terhadap Nilai Eror pada Model *Bagging Regression Tree* untuk Rute Jakarta-Manado

Nilai eror *out-of-bag* (OOB) akan dihitung menggunakan RMSE. Terlihat bahwa nilai eror mulai stabil saat jumlah model pohon sebanyak 25. Model tidak membaik secara signifikan dengan penambahan jumlah model pohon, sehingga model *Bagging Regression Tree* untuk rute Jakarta-Manado akan dibuat dengan hanya menggunakan 25 model pohon. model *Bagging Regression Tree* untuk rute Jakarta-Manado beserta hasil *10-fold* CV untuk model ini adalah sebagai berikut.

```
> bagged_ml
Bagged CART

43597 samples
   10 predictor

No pre-processing
Resampling: Cross-validated (10 fold)
Summary of sample sizes: 39239, 39238, 39237, 39237, 39237, 39237,
    ...
Resampling results:

  RMSE      Rsquared     MAE
  317942.2  0.8148509    244818.4
```

Berdasarkan nilai RMSE, model ini memiliki kesalahan prediksi sebesar Rp317.942,20 secara rata-rata, sedangkan berdasarkan nilai MAE, model ini memiliki kesalahan prediksi sebesar Rp244.818,40 secara rata-rata. Nilai R^2 menunjukkan bahwa 81,49% variabilitas pada variabel respons dapat dijelaskan oleh model ini. Tingkat *variable importance* ditunjukkan pada Gambar 4.33.



Gambar 4.33: Urutan *Variable Importance* pada Model *Bagging Regression Tree* untuk Rute Jakarta-Manado

Variabel `duration` adalah prediktor yang memberikan dampak terbesar terhadap SSE sehingga menjadi variabel terpenting. Selain variabel `duration`, variabel `departure` dan `arrival` juga memberikan dampak yang cukup besar pada SSE.

4.5.2.2 Performa Model

Hasil *training error* dan *testing error* model *Bagging Regression Tree* untuk rute Jakarta-Manado ditunjukkan pada Tabel 4.12. Nilai *training error* dan *testing error* untuk model ini tidak saling jauh berbeda.

Tabel 4.12: Performa Model *Bagging Regression Tree* untuk Rute Jakarta-Manado

Performa	*Training Error*	*Testing Error*
MAE	244.531,6	247.941,0
MAPE	8,06%	8,13%
RMSE	317.672,9	322.254,0
R^2	81,53%	81,08%

Berdasarkan nilai MAE dan MAPE, model ini cenderung memiliki kesalahan prediksi sebesar Rp247.941,00 atau 8,13% dari nilai sebenarnya secara rata-rata, sedangkan berdasarkan nilai RMSE, model ini cenderung memiliki kesalahan prediksi sebesar Rp322.254,00 secara rata-rata. Nilai R^2 menunjukkan bahwa 81,08% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.5.3 *Random Forest*

4.5.3.1 Model *Random Forest* Awal

Ditentukan nilai *default* untuk ketiga parameter pada model *Random Forest* yaitu `mtry=5`, `ntree=500`, dan `maxnodes=10`. Model *Random Forest* awal untuk rute Jakarta-Manado beserta hasil *10-fold* CV untuk model ini adalah sebagai berikut.

```
> print(rf_default)
Random Forest

43597 samples
   10 predictor

No pre-processing
Resampling: Cross-validated (5 fold)
Summary of sample sizes: 34880, 34876, 34878, 34877, 34877
Resampling results:

  RMSE      Rsquared    MAE
  483940.9  0.6883878   344294

Tuning parameter 'mtry' was held constant at a value of 5
```

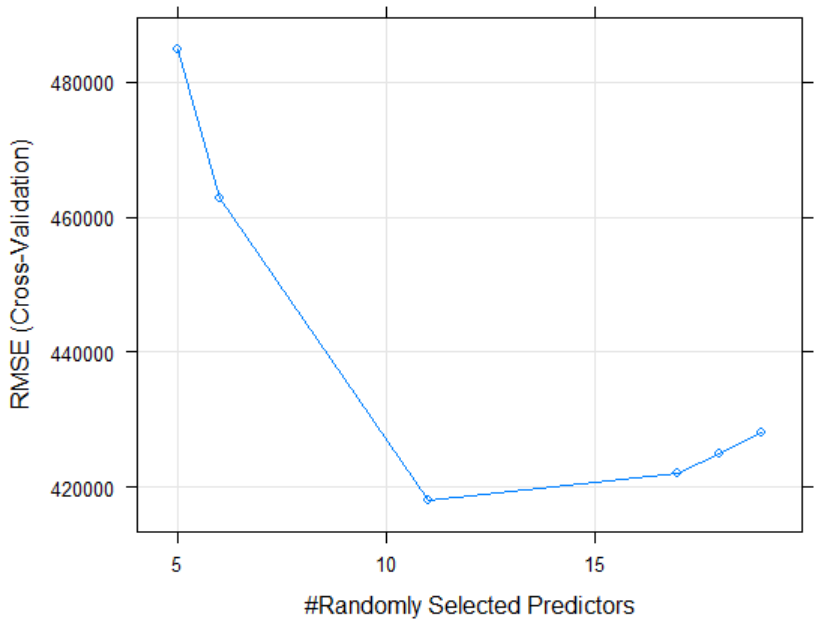
Berdasarkan nilai RMSE, model ini memiliki kesalahan prediksi sebesar Rp483.940,90 secara rata-rata, sedangkan berdasarkan nilai MAE, model ini memiliki kesalahan prediksi sebesar Rp344.294,80 secara rata-rata. Nilai R^2 menunjukkan bahwa 68,83% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.5.3.2 Model *Tuned Random Forest*

Parameter tuning akan dilakukan terhadap ketiga parameter yaitu `mtry`, `ntree`, dan `maxnodes`. Setiap nilai parameter akan dibandingkan nilai erornya menggunakan *5-fold* CV. Hasil *parameter tuning* terhadap `mtry` adalah sebagai berikut.

```
> print (rf_random)
Random Forest

43597 samples
   10 predictor

No pre-processing
Resampling: cross-validated (5 fold)

Sunmary of sanple sizes: 34878, 34877, 34877, 34880, 34876
Resampling results across tuning paraneters:

  mtry  RMSE        Rsquared    MAE
  5     484780.1    0.6916179   345470.5
  6     462822.1    0.7036381   328701.0
  11    418109.3    0.7167562   293464.8
  17    422035.4    0.6800593   295906.0
  18    424955.1    0.6735668   297834.7
  19    428096.5    0.6669783   300669.5

RMSE was used to select the optimal model using the smallest
    value. The final value used for the model was mtry = 11.
```

Parameter tuning terhadap `mtry` dilakukan menggunakan *random search*. Dipilih 10 nilai `mtry` secara acak dan untuk setiap nilai `mtry` akan dicari nilai erornya menggunakan *5-fold* CV. Terlihat pada Gambar 4.34 bahwa `mtry=11` adalah nilai yang paling optimal.



Gambar 4.34: *Plot* Hasil *Parameter Tuning* terhadap Parameter *mtry* pada Model *Random Forest* untuk Rute Jakarta-Manado

Parameter tuning terhadap `ntree` dilakukan secara manual. Terdapat tiga nilai `ntree` yang akan dibandingkan yaitu 200, 500, dan 1000. Dari ketiga nilai tersebut, akan dicari yang paling optimal. Pada Tabel 4.13 ditampilkan hasil *parameter tuning* terhadap `ntree`. Terlihat bahwa `ntree=1000` adalah nilai yang paling optimal.

Tabel 4.13: *Parameter Tuning* terhadap Parameter *ntree* pada Model *Random Forest* untuk Rute Jakarta-Manado

	MAE	RMSE	*R-Squared*
200	293.398,5	417.190,1	0,7152022
500	292.936,6	416.846,2	0,7166702
1000	292.487,5	416.453,4	0,7180172

Parameter tuning terhadap `maxnodes` dilakukan secara manual. Terdapat tiga nilai `maxnodes` yang akan dibandingkan yaitu 5, 10, 20, 30, 40, 50. Dari keenam nilai tersebut, akan dicari yang paling optimal. Pada Tabel 4.14 ditampilkan hasil *parameter tuning* terhadap `maxnodes`. Terlihat bahwa `maxnodes=50` adalah nilai yang paling optimal.

Tabel 4.14: *Parameter Tuning* terhadap Parameter *maxnodes* pada Model *Random Forest* untuk Rute Jakarta-Manado

	MAE	RMSE	*R-Squared*
5	371.635,1	521.942,3	0,5471632
10	292.487,5	416.453,4	0,7180172
20	245.736,1	346.152,6	0,7958762
30	223.311,3	304.723,2	0,8393045
40	216.789,4	294.878,8	0,8472355
50	208.023,9	278.516,7	0,8622235

Dari hasil *parameter tuning*, didapatkan nilai yang paling optimal untuk model *Random Forest* yaitu `mtry=11`, `ntree=1000`, dan `maxnodes=50`. Model *Tuned Random Forest* untuk rute Jakarta-Manado adalah sebagai berikut.

```
> rf_opt
Random Forest

43357 samples
   10 predictor

No pre-processing
Resampling: Cross-validated (5 fold)
Summary of sample sizes: 34880, 34876, 34878, 34877, 34877
Resampling results:

  RMSE      Rsquared    MAE
  278020.2  0.8625932   207933.6

Tuning parameter 'mtry' was held constant at a value of 11
```

Berdasarkan nilai RMSE, model ini memiliki kesalahan prediksi sebesar Rp278.020,20 secara rata-rata, sedangkan berdasarkan nilai MAE, model ini memiliki kesalahan prediksi sebesar Rp207.933,60 secara rata-rata. Nilai R^2 menunjukkan bahwa 86,26% variabilitas pada variabel respons dapat dijelaskan oleh model ini. Ketiga nilai tersebut jauh lebih baik daripada model *Random Forest* awal yang dijelaskan pada bagian 4.5.3.1, sehingga model akhir yang digunakan adalah model *Tuned Random Forest*.

4.5.3.3 Performa Model

Hasil *training error* dan *testing error* model *Random Forest* untuk rute Jakarta-Manado ditunjukkan pada Tabel 4.15. Nilai *training error* dan *testing error* untuk model ini tidak saling jauh berbeda.

Tabel 4.15: Performa Model *Random Forest* untuk Rute Jakarta-Manado

Performa	*Training Error*	*Testing Error*
MAE	207.099,2	209.333,0
MAPE	6.81%	6,85%
RMSE	277.298,5	279.537,7
R^2	86,34%	86,04%

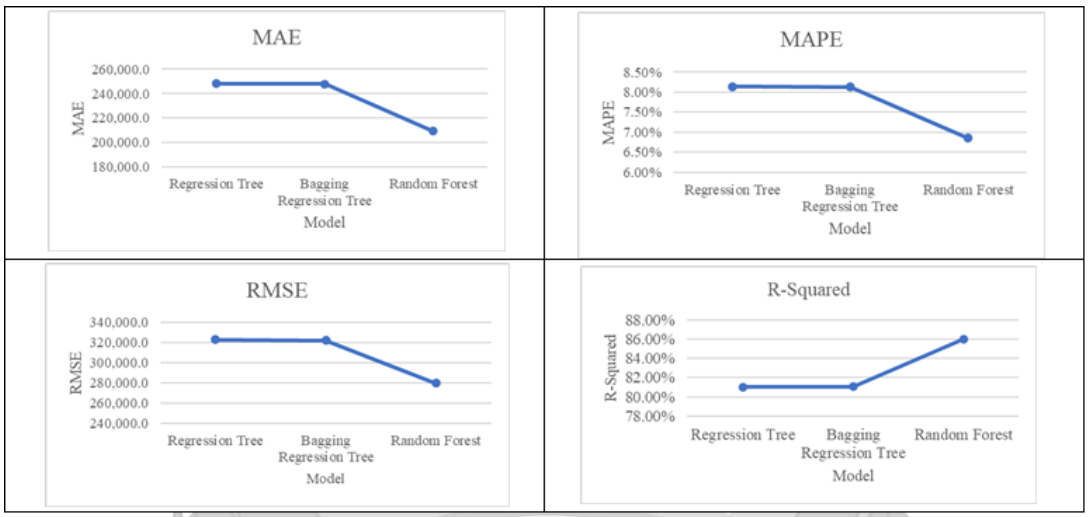
Berdasarkan nilai MAE dan MAPE, model ini cenderung memiliki kesalahan prediksi sebesar Rp209.333,00 atau 6,85% dari nilai sebenarnya secara rata-rata, sedangkan berdasarkan nilai RMSE, model ini cenderung memiliki kesalahan prediksi sebesar Rp279.537,70 secara rata-rata. Nilai R^2 menunjukkan bahwa 86,04% variabilitas pada variabel respons dapat dijelaskan oleh model ini.

4.5.4 Evaluasi Performa Seluruh Model

Performa tiap model akan dibandingkan melalui kemampuan tiap model dalam melakukan prediksi. Nilai *testing error* setiap model ditunjukkan pada tabel 4.16 dan divisualisasikan pada Gambar 4.35.

Tabel 4.16: Performa Model untuk Rute Jakarta-Manado

Model	MAE	MAPE	RMSE	R^2
Regression Tree	248.163,6	8,14%	322.838,7	81,01%
Bagging Regression Tree	247.941,0	8,13%	322.254,0	81,08%
Random Forest	209.333,0	6,85%	279,537.7	86,04%



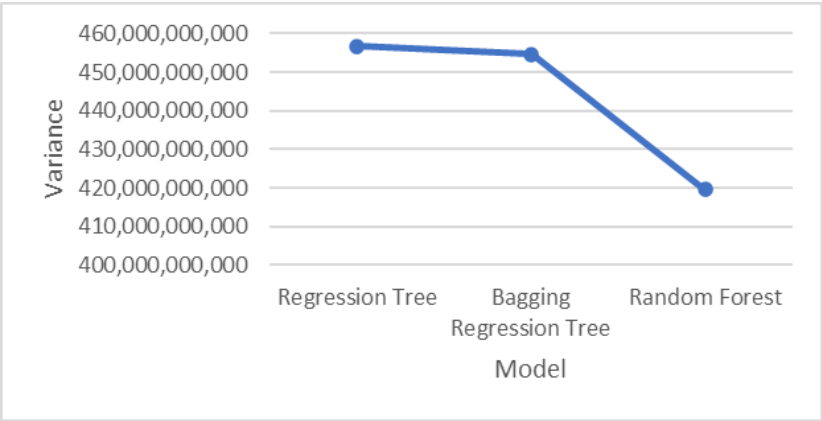
Gambar 4.35: Visualisasi Performa Model untuk Rute Jakarta-Manado

Model *Random Forest* memiliki nilai MAE, MAPE, dan RMSE terendah. Model *Random Forest* juga memiliki nilai R^2 tertinggi, sehingga model terbaik untuk rute Jakarta-Manado adalah model *Random Forest*. Model *Regression Tree* dan *Bagging Regression Tree* tidak berbeda secara signifikan, namun model *Regression Tree* memiliki nilai MAE, MAPE, RMSE, dan R^2 yang lebih baik.

Nilai bias dan varians ditunjukkan pada Tabel 4.17. Nilai varians pada setiap model jauh lebih besar dibandingkan nilai biasnya karena ketiga model tersebut merupakan model nonparametrik yang fleksibel. Model *Bagging Regression Tree* dan *Random Forest* dilakukan untuk mengurangi varians dibandingkan model *Regression Tree*. Berdasarkan hasil yang didapatkan, kedua model tersebut dapat mengurangi varians seperti yang ditunjukkan pada Gambar 4.36

Tabel 4.17: Nilai Bias dan Varians Model Prediksi untuk Rute Jakarta-Manado

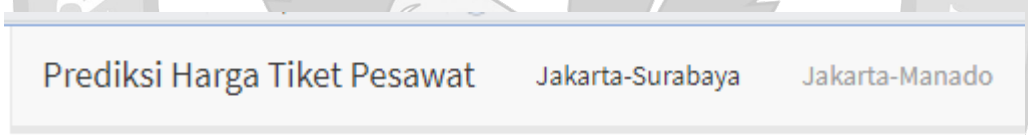
Model	Bias	Varians
Regression Tree	558.622,5	456.779.773.820
Bagging Regression Tree	1.202.676,0	454.643.786.412
Random Forest	633.271,5	419.486.542.605



Gambar 4.36: Nilai Varians Model Prediksi untuk Rute Jakarta-Manado

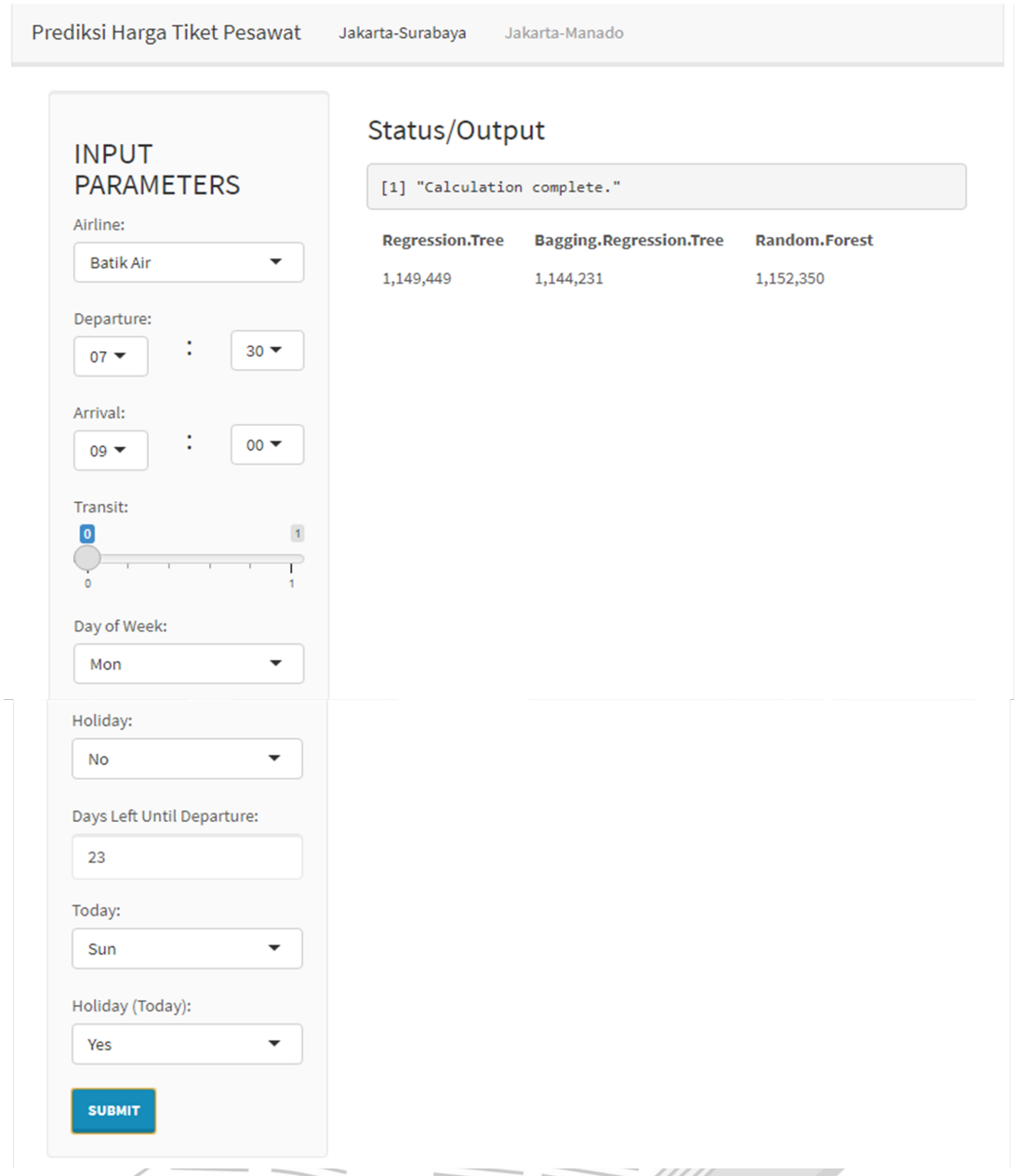
4.6 *Web App* Prediksi Harga Tiket Pesawat Menggunakan *R Shiny*

Ketiga model dari setiap rute akan digunakan untuk membangun sebuah *web app* menggunakan *R Shiny*. Pada Gambar 4.37 ditunjukkan *navigation bar* pada *web app* "Prediksi Harga Tiket Pesawat". *Navigation bar* tersebut dapat digunakan untuk berpindah halaman antara halaman "Jakarta-Surabaya" dan halaman "Jakarta-Manado".



Gambar 4.37: *Navigation Bar Web App* Prediksi Harga Tiket Pesawat

Pada Gambar 4.38 ditunjukkan tampilan *web app* Prediksi Harga Tiket Pesawat secara lengkap. Pada setiap halaman, terdapat *side bar panel* dan *main panel*. Pada *side bar panel* disediakan tempat untuk melakukan *input* parameter seperti jenis maskapai, jam keberangkatan, jam kedatangan, transit, dan sebagainya. Pada *main panel* ditampilkan status dari aplikasi, yaitu *"Server is ready for calculation."* jika model siap digunakan dan *"Calculation complete."* jika hasil prediksi sudah didapatkan. Pada *main panel* juga ditampilkan hasil prediksi harga tiket pesawat menggunakan ketiga model yaitu *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*.



Gambar 4.38: *Web App* Prediksi Harga Tiket Pesawat

BAB V

PENUTUP

5.1 Kesimpulan

Prediksi harga tiket pesawat maskapai penerbangan di Indonesia pada rute Jakarta-Surabaya dan Jakarta-Manado berhasil dilakukan menggunakan ketiga model ML yang telah ditentukan yaitu *Regression Tree*, *Bagging Regression Tree*, dan *Random Forest*. Evaluasi terhadap model yang telah didapatkan tersebut dilakukan dengan membandingkan nilai MAE, MAPE, RMSE, dan R^2. Berdasarkan Tabel 2.1, hasil prediksi harga tiket pesawat dari setiap model prediksi untuk kedua rute yang telah diperoleh cukup akurat dengan nilai MAPE di bawah 20%. Model *Random Forest* memiliki tingkat kesalahan prediksi terkecil. Oleh karena itu, model prediksi terbaik untuk kedua rute tersebut adalah *Random Forest*.

5.2 Saran

Berdasarkan hasil penelitian, saran yang dapat diberikan adalah sebagai berikut.

1. Pada penelitian ini, model hanya dapat memprediksi harga tiket pesawat pada dua rute yaitu Jakarta-Surabaya dan Jakarta-Manado. Penelitian lebih lanjut dapat dilakukan untuk mengembangkan model tersebut agar bisa memprediksi harga tiket pesawat pada rute lain juga.

2. Penelitian ini hanya menggunakan data penerbangan selama tiga bulan yaitu dari bulan Juni hingga Agustus. Penelitian selanjutnya dapat menggunakan data selama setahun dibandingkan hanya tiga bulan.

DAFTAR PUSTAKA

[1] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pages 42–47. IEEE, 2013.

[2] Lidong Wang and Cheryl Ann Alexander. Machine learning in big data. *International Journal of Mathematical, Engineering and Management Sciences*, 1(2):52–61, 2016.

[3] Linda A Clark and Daryl Pregibon. Tree-based models. In *Statistical models in S*, pages 377–419. Routledge, 2017.

[4] Anantha M Prasad, Louis R Iverson, and Andy Liaw. Newer classification and regression rree techniques: Bagging and random forests for ecological prediction. *Ecosystems*, 9(2):181–199, 2006.

[5] Siti Fatimah. *Pengantar Transportasi*. Myria Publisher, 2019.

[6] Luo Li and Ji-Hua Peng. Dynamic pricing model for airline revenue management under competition. *Systems Engineering-Theory & Practice*, 27(11):15–25, 2007.

[7] Terry Therneau, Beth Atkinson, and Brian Ripley. rpart: Recursive partitioning and regression trees. *R package version*, 4:1–9, 2015.

[8] Max Kuhn. Building predictive models in r using the caret package. *Journal of Statistical Software, Articles*, 28(5):1–26, 2008.

[9] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. Machine learning from theory to algorithms: An overview. In *Journal of physics: conference series*, volume 1142, page 012012. IOP Publishing, 2018.

[10] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 1967.

[11] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 112. Springer, 2013.

[12] Mei-Hung Chiu, Yuh-Ru Yu, Hongming Leonard Liaw, and Lin Chun-Hao. The use of facial micro-expression state and tree-forest model for predicting conceptual-conflict based conceptual change. *Chapter Title & Authors Page*, 184, 2016.

[13] Jason Brownlee. Master machine learning algorithms: Discover how they work and implement them from scratch, 2016. *URL https://books. google. ca/books*.

[14] Viola Citra Dewi. *Using Decision Tree-Based Data Mining to Predict Types of Apparels*. PhD thesis, Universitas Pelita Harapan, 2021.

[15] Radwa Elshawi, Mohamed Maher, and Sherif Sakr. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*, 2019.

[16] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

[17] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[18] Samantha Sanders and Christophe Giraud-Carrier. Informing the use of hyperparameter optimization through metalearning. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1051–1056. IEEE, 2017.

[19] DG Mayer and DG Butler. Statistical validation. *Ecological Modelling*, 68(1-2):21–32, 1993.

[20] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific Model Development Discussions*, 7(1):1525–1534, 2014.

[21] Arnaud De Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016.

[22] Colin David Lewis. *Industrial and business forecasting methods: A practical guide to exponential smoothing and curve fitting*. Butterworth-Heinemann, 1982.

[23] Alessandro Di Bucchianico. Coefficient of determination (r 2). *Encyclopedia of Statistics in Quality and Reliability*, 1, 2008.

[24] Konstantinos Tziridis, Th Kalampokas, George A Papakostas, and Kostas I Diamantaras. Airfare prices prediction using machine learning techniques. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1036–1039. IEEE, 2017.

[25] Carolin Strobl, James Malley, and Gerhard Tutz. An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, 14(4):323, 2009.

[26] Camino González, José Mira-McWilliams, and Isabel Juárez. Important variable assessment and electricity price forecasting based on regression tree models: Classification and regression trees, bagging and random forests. *IET Generation, Transmission & Distribution*, 9(11):1120–1128, 2015.

Lampiran A

Data

A.1 Data Penerbangan Rute Jakarta-Surabaya

Tabel A.1: Data Penerbangan Rute Jakarta-Surabaya

airline	departure	arrival	transit	day_ of_ week	holiday	days_left_ until_departure	duration (jam)	scrape_ day	scrape_ holiday	price (rupiah)
Batik Air	0.17	0.23	0	1	TRUE	59	1.5	5	FALSE	927200
Super Air Jet	0.61	0.68	0	7	TRUE	51	1.5	5	FALSE	816100
Citilink	0.23	0.29	0	5	FALSE	35	1.5	5	FALSE	870451
Citilink	0.50	0.56	0	5	FALSE	35	1.5	5	FALSE	925951
Lion Air	0.73	0.79	0	4	FALSE	27	1.5	5	FALSE	801100
Batik Air	0.38	0.44	0	5	FALSE	21	1.5	5	FALSE	927200
Batik Air	0.33	0.40	0	3	FALSE	26	1.5	5	FALSE	992700
Super Air Jet	0.24	0.30	0	2	FALSE	32	1.5	5	FALSE	816100
Batik Air	0.21	0.27	0	4	TRUE	69	1.5	5	FALSE	927200
Citilink	0.19	0.25	0	6	FALSE	43	1.5	5	FALSE	980341
Batik Air	0.29	0.35	0	1	TRUE	59	1.5	5	FALSE	919200
Batik Air	0.17	0.23	0	7	FALSE	65	1.5	5	FALSE	927200
Super Air Jet	0.78	0.84	0	6	FALSE	29	1.5	5	FALSE	871600
Batik Air	0.25	0.31	0	7	FALSE	65	1.5	5	FALSE	927200
Batik Air	0.21	0.27	0	4	FALSE	13	1.5	5	FALSE	862900

airline	departure	arrival	transit	day_ of_ week	holiday	days_left_ until_departure	duration (jam)	scrape_ day	scrape_ holiday	price (rupiah)
Batik Air	0.58	0.65	0	7	FALSE	72	1.5	5	FALSE	927200
Lion Air	0.52	0.58	0	1	TRUE	38	1.5	5	FALSE	801100
Lion Air	0.60	0.67	0	5	FALSE	14	1.5	5	FALSE	801100
Batik Air	0.33	0.40	0	6	FALSE	57	1.5	5	FALSE	927200
Batik Air	0.63	0.69	0	4	FALSE	55	1.5	5	FALSE	927200
Batik Air	0.38	0.44	0	2	FALSE	67	1.5	5	FALSE	927200
Super Air Jet	0.78	0.84	0	5	FALSE	49	1.5	5	FALSE	816100
Nam Air	0.25	0.51	1	2	FALSE	32	6.25	5	FALSE	2238768
Batik Air	0.17	0.23	0	7	TRUE	51	1.5	5	FALSE	927200
Garuda Indonesia	0.31	0.38	0	3	FALSE	61	1.666667	5	FALSE	1122900
Lion Air	0.15	0.22	0	1	TRUE	17	1.5	5	FALSE	801100
Batik Air	0.33	0.40	0	6	FALSE	50	1.5	5	FALSE	992700
Citilink	0.23	0.29	0	7	FALSE	23	1.5	5	FALSE	870451
Citilink	0.50	0.56	0	2	FALSE	18	1.5	5	FALSE	980341
Batik Air	0.21	0.27	0	6	FALSE	15	1.5	5	FALSE	927200
Batik Air	0.21	0.27	0	1	TRUE	31	1.5	5	FALSE	862900
Lion Air	0.65	0.71	0	3	FALSE	12	1.5	5	FALSE	801100
Super Air Jet	0.61	0.68	0	1	TRUE	24	1.5	5	FALSE	871600
Nam Air	0.25	0.51	1	6	FALSE	50	6.25	5	FALSE	2238768
Garuda Indonesia	0.68	0.75	0	6	FALSE	22	1.666667	5	FALSE	1544700

airline	departure	arrival	transit	day_ of_ week	holiday	days_left_ until_departure	duration (jam)	scrape_ day	scrape_ holiday	price (rupiah)
Lion Air	0.08	0.40	1	4	FALSE	27	7.666667	5	FALSE	2248400
Lion Air	0.65	0.71	0	7	FALSE	16	1.5	5	FALSE	801100
Super Air Jet	0.24	0.30	0	4	FALSE	6	1.5	5	FALSE	816100
Citilink	0.50	0.56	0	7	FALSE	44	1.5	5	FALSE	870451
Super Air Jet	0.70	0.76	0	4	FALSE	6	1.5	5	FALSE	871600
Garuda Indonesia	0.67	0.74	0	4	FALSE	41	1.666667	5	FALSE	1122900
Batik Air	0.75	0.81	0	7	TRUE	51	1.5	5	FALSE	927200
Citilink	0.50	0.56	0	5	FALSE	56	1.5	5	FALSE	1310011
Batik Air	0.29	0.35	0	7	FALSE	79	1.5	5	FALSE	919200
Citilink	0.23	0.29	0	2	FALSE	60	1.5	5	FALSE	870451
Garuda Indonesia	0.72	0.79	0	1	TRUE	59	1.666667	5	FALSE	1122900
Lion Air	0.23	0.29	0	2	FALSE	60	1.5	5	FALSE	801100
Lion Air	0.52	0.58	0	3	FALSE	5	1.5	5	FALSE	911000
Garuda Indonesia	0.35	0.42	0	1	TRUE	3	1.666667	5	FALSE	1544700
Citilink	0.62	0.69	0	4	TRUE	69	1.666667	5	FALSE	870451

A.2 Data Penerbangan Rute Jakarta-Manado

Tabel A.2: Data Penerbangan Rute Jakarta-Manado

airline	departure	arrival	transit	day_ of_ week	holiday	days_left_ until_departure	duration (jam)	scrape_ day	scrape_ holiday	price (rupiah)
Batik Air	0.11	0.53	1	5	FALSE	63	10.08333	5	FALSE	3409400
Batik Air	0.10	0.59	1	4	FALSE	83	11.66667	5	FALSE	2819100
Batik Air	0.10	0.41	1	2	FALSE	60	7.416667	5	FALSE	2483900
Citilink	0.10	0.25	0	3	FALSE	82	3.416667	5	FALSE	1945544
Batik Air	0.11	0.53	1	7	FALSE	44	10.08333	5	FALSE	3409400
Lion Air	0.17	0.56	2	2	FALSE	53	9.416667	5	FALSE	3757200
Batik Air	0.25	0.40	0	7	FALSE	79	3.5	5	FALSE	1964900
Garuda Indonesia	0.11	0.26	0	5	FALSE	42	3.416667	5	FALSE	2454900
Lion Air	0.08	0.41	1	7	FALSE	2	7.916667	5	FALSE	2255800
Lion Air	0.15	0.43	1	5	FALSE	77	6.75	5	FALSE	2249500
Citilink	0.10	0.25	0	6	FALSE	64	3.416667	5	FALSE	2190854
Lion Air	0.38	0.59	1	4	FALSE	83	5	5	FALSE	2677600
Batik Air	0.44	0.59	0	7	FALSE	58	3.5	5	FALSE	2109200
Batik Air	0.11	0.53	1	3	FALSE	40	10.08333	5	FALSE	3409400
Batik Air	0.08	0.41	1	2	FALSE	18	7.916667	5	FALSE	2549400

airline	departure	arrival	transit	day_ of_ week	holiday	days_left_ until_departure	duration (jam)	scrape_ day	scrape_ holiday	price (rupiah)
Batik Air	0.10	0.41	1	6	FALSE	1	7.416667	5	FALSE	2481700
Batik Air	0.10	0.49	2	1	TRUE	24	9.25	5	FALSE	3680200
Batik Air	0.10	0.59	1	4	FALSE	55	11.66667	5	FALSE	2819100
Batik Air	0.10	0.25	0	4	FALSE	76	3.5	5	FALSE	1964900
Lion Air	0.08	0.49	2	1	TRUE	45	9.75	5	FALSE	3242200
Batik Air	0.44	0.59	0	1	TRUE	45	3.5	5	FALSE	2398900
Batik Air	0.15	0.41	1	4	FALSE	41	6.416667	5	FALSE	2449500
Lion Air	0.23	0.43	1	1	TRUE	38	4.916667	5	FALSE	2052400
Batik Air	0.44	0.59	0	6	FALSE	22	3.5	5	FALSE	2398900
Lion Air	0.38	0.59	1	7	FALSE	58	5	5	FALSE	2677600
Batik Air	0.10	0.41	1	1	TRUE	3	7.416667	5	FALSE	2112100
Garuda Indonesia	0.11	0.26	0	6	FALSE	64	3.416667	5	FALSE	2299500
Lion Air	0.15	0.43	1	4	FALSE	83	6.75	5	FALSE	2249500
Lion Air	0.15	0.41	1	3	FALSE	12	6.416667	5	FALSE	2169200
Lion Air	0.23	0.43	1	7	FALSE	9	4.916667	5	FALSE	2052400
Batik Air	0.11	0.56	1	1	TRUE	31	10.75	5	FALSE	3409400
Batik Air	0.25	0.40	0	5	FALSE	35	3.5	5	FALSE	2254600
Batik Air	0.11	0.56	1	3	FALSE	61	10.75	5	FALSE	3409400
Lion Air	0.15	0.56	2	2	FALSE	4	9.916667	5	FALSE	3486400
Batik Air	0.11	0.38	1	6	FALSE	57	6.416667	5	FALSE	3409400

airline	departure	arrival	transit	day_ of_ week	holiday	days_left_ until_departure	duration (jam)	scrape_ day	scrape_ holiday	price (rupiah)
Garuda Indonesia	0.11	0.26	0	3	FALSE	26	3.416667	5	FALSE	3165300
Lion Air	0.17	0.41	1	7	FALSE	65	5.916667	5	FALSE	2255800
Citilink	0.10	0.25	0	3	FALSE	47	3.416667	5	FALSE	1945544
Batik Air	0.11	0.53	1	7	FALSE	9	10.08333	5	FALSE	3887800
Batik Air	0.44	0.59	0	1	TRUE	38	3.5	5	FALSE	2398900
Lion Air	0.17	0.53	2	7	TRUE	30	8.75	5	FALSE	3757200
Lion Air	0.17	0.56	2	2	FALSE	74	9.416667	5	FALSE	3757200
Lion Air	0.38	0.59	1	3	FALSE	54	5	5	FALSE	2677600
Lion Air	0.17	0.41	1	6	FALSE	50	5.916667	5	FALSE	2323500
Lion Air	0.17	0.53	2	5	FALSE	56	8.75	5	FALSE	3757200
Batik Air	0.11	0.53	1	1	TRUE	66	10.08333	5	FALSE	3569200
Batik Air	0.11	0.38	1	7	FALSE	58	6.416667	5	FALSE	3409400
Lion Air	0.17	0.59	1	4	FALSE	83	10.16667	5	FALSE	2591000
Lion Air	0.23	0.43	1	7	TRUE	30	4.916667	5	FALSE	2052400
Lion Air	0.15	0.41	1	2	FALSE	18	6.416667	5	FALSE	2437800