ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 14, Nomor 2, Tahun 2025, Halaman 269 - 279
Online di: https://ejournal3.undip.ac.id/index.php/gaussian/

PENERAPAN MODEL REGRESI *RANDOM FOREST* UNTUK PREDIKSI HARGA LAPTOP BERDASARKAN FITUR LAPTOP

Iftahli Nurol Ilmi[1*], Mustafid[2], Suparti[3]
[1,2,3] Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro
***e-mail**: iftahli13@gmail.com

DOI: 10.14710/j.gauss.14.2.269-279

Article Info:
Received: 2024-07-17
Accepted: 2025-09-11
Available Online: 2025-09-17

Keywords:
Laptop Prices; Laptop Features; Random Forest Regression; Random Search CV; bootstrap

Abstract: The increasing use of computer science systems in various fields of work is driving the growth of laptop. There are many choices of laptop products with different specifications. Laptop price predictions can help consumers to find out the price range of laptops according to the laptop features they want. This study aims to apply the random forest regression method to predict laptop price based on laptop features. The data is splitted into 2 parts, 1130 data as training set and 283 data as testing set. Hyperparameters tuning was performed using random search CV with 10 fold cross-validation to find the combination of hyperparameters that produced the optimal model. The best hyperparameters obtained to build a random forest regression model are ntree = 400, mtry = 2, and nodesize = 2. The MAPE value of the testing set for the random forest regression model is 14,3% which indicates the performance of the model has good forecasting ability. The results of the analysis show that the random forest regression method can be applied to predict laptop prices based on laptop features. Based on the variable importance, the variable that has the greatest contribution to the laptop price prediction results is RAM with VI = 0,359.

1. PENDAHULUAN

Peningkatan penggunaan sistem ilmu komputer pada berbagai bidang pekerjaan mendorong pertumbuhan pasar PC. Lembaga riset IDC (2022) mencatat pasar PC tradisional (desktop, laptop, dan *workstation*) di kawasan Asia/Pasifik mengalami peningkatan 15,9% YoY pada tahun 2021. IDC (2022) juga mencatat laptop menjadi kategori pendorong akan pertumbuhan pasar produk PC. Produk laptop di pasaran memiliki banyak pilihan dengan spesifikasi yang berbeda-beda sehingga konsumen dapat memilih sesuai kebutuhan dan anggaran. Prediksi harga laptop dapat membantu calon konsumen untuk mengetahui kisaran harga laptop sesuai fitur laptop yang diinginkan.

Beberapa penerapan metode statistika telah banyak diperkenalkan untuk melakukan prediksi suatu variabel respon berdasarkan faktor yang memengaruhinya. Metode-metode yang dikembangkan dalam ilmu komputer dan statistika bermunculan, salah satunya adalah metode *random forest*.

Random forest diusulkan oleh Breiman pada tahun 2001 dan merupakan pengembangan dari metode pohon keputusan yang rawan *overfitting*. Metode *random forest* dapat digunakan untuk kasus regresi dan klasifikasi. Beberapa penelitian mengenai penerapan metode regresi *random forest* pernah dilakukan di antaranya penelitian oleh Pal *et al*. (2018) yang memprediksi harga mobil bekas berdasarkan fitur harga, kilometer, merek, dan jenis kendaraan mendapatkan hasil bahwa metode regresi *random forest* dapat digunakan untuk memprediksi harga dengan akurasi yang baik.

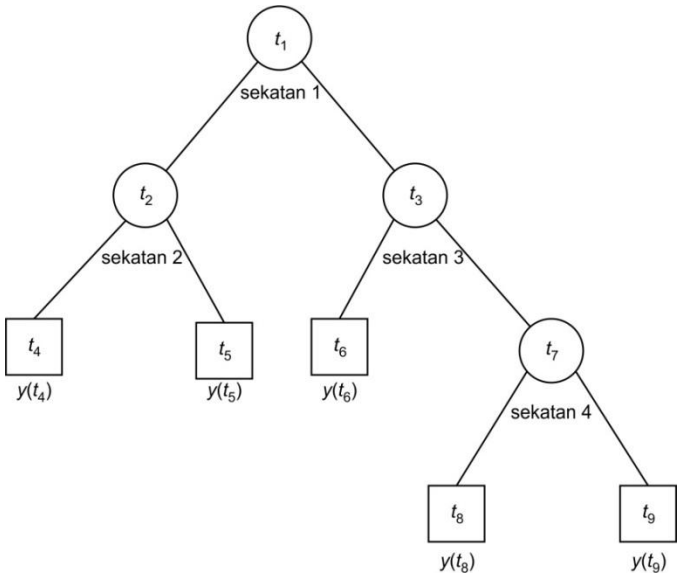
Metode regresi *random forest* diterapkan dalam penelitian ini untuk membuat model prediksi harga laptop berdasarkan fitur laptop. Penyetelan *hyperparameter* untuk mendapat model regresi *random forest* yang optimal dilakukan menggunakan *random search CV*.

2. TINJAUAN PUSTAKA

Random forest adalah metode *ensemble* berbasis pohon keputusan yang dikembangkan untuk mengatasi kelemahan metode pohon keputusan. *Random forest* terdiri dari sejumlah besar pohon keputusan yang lemah, yang ditumbuhkan secara paralel untuk mengurangi bias dan varians model pada saat yang sama (Breiman, 2001). Metode ini menerapkan teknik *bagging* dan metode pemilihan fitur secara acak (Breiman, 2001).

Bagging atau *bootstrap aggregating* merupakan metode yang manggabungkan beberapa algoritma pembelajaran. Jatmiko (2019) menyatakan ide dasar dari metode *bagging* adalah menggunakan *resampling* acak dengan pengembalian pada dataset awal yang kemudian diperoleh suatu dataset baru. Dataset baru digunakan untuk membangkitkan banyak versi pohon keputusan. Pohon keputusan dari setiap versi kemudian digabungkan untuk mendapatkan prediksi akhir.

Chen dan Howard (2015) menjelaskan bahwa pohon keputusan atau *decision tree* adalah salah satu metode pembelajaran mesin populer yang mempunyai fungsi klasifikasi dan regresi. Fitur data dalam metode pohon keputusan disebut sebagai variabel prediktor sedangkan kelas yang akan dipetakan adalah variabel target. Variabel target untuk masalah regresi adalah kontinu (Breiman *et al*., 1984). Struktur pohon regresi dapat digambarkan pada Gambar 1 (Breiman *et al*., 1984).



Gambar 1. Struktur Pohon Regresi

Pohon regresi dapat ditulis sebagai semacam model aditif dengan persamaan (1) (Hastie *et al*., 2009),

$$\hat{f}(x) = \sum_{i=1}^{l} k_i \times I(x \in D_i) \qquad (1)$$

dengan k_i adalah konstanta, $I(\cdot)$ adalah fungsi indikator yang mengembalikan 1 jika argumen benar dan 0 jika sebaliknya, dan D_i adalah partisi penyekatan dari data D sehingga $\bigcup_{i=1}^{l} D_i = D$ dan $\bigcap_{i=1}^{l} D_i = \phi$. Misal j adalah variabel penyekat dan s adalah titik penyekat, maka persamaan dari kedua bagian yang tersekat tersebut seperti persamaan (2).

$$D_1(j,s) = \{X \mid X_j \le s\} \text{ dan } D_2(j,s) = \{X \mid X_j > s\} \qquad (2)$$

Aproksimasi fungsi regresi *random forest* sebagai algoritma *ensemble* berbasis pohon untuk sampel pelatihan $D_N = (X, Y)$ dirumuskan dengan persamaan (3) (Scornet, 2016),

$$\hat{f}_{T,N}(x) = \frac{1}{T}\sum_{i=1}^{T}\hat{f}_N(x, \Theta_i) \quad (3)$$

dengan T adalah banyak pohon regresi acak di hutan, N adalah ukuran data, $\hat{f}_N(x, \Theta_i)$ menunjukkan nilai prediksi titik x untuk pohon ke-i tunggal (yaitu dibangun menggunakan sampel *bootstrap*), Θ_i mengkarakterisasi pohon ke-i dalam hal variabel penyekatan, titik penyekat pada setiap simpul, dan nilai simpul terminal, serta X dan Y adalah variabel prediktor dan variabel respon.

Algoritma regresi *random forest* terdapat empat langkah sebagai berikut (Xu *et al*., 2019):

1. Gunakan teknik *bootstrap* untuk menghasilkan *ntree* subset sampel dari dataset pelatihan asli, dengan *ntree* adalah banyak pohon yang akan tumbuh.
2. Tumbuhkan pohon regresi pada setiap sampel *bootstrap*, gambarkan secara acak subset yang berisi variabel prediktor *mtry* pada setiap simpul yang membelah dan tentukan pemisahan optimal berdasarkan subset variabel ini saja. Proses ini dilakukan secara rekursif sampai kriteria berhenti (*nodesize*) tercapai.
3. Dapatkan prediksi regresi dari *ntree* pohon keputusan. Untuk setiap pohon individu, prediksi adalah rata-rata dari nilai variabel respon pada simpul daun yang sesuai.
4. Hitung prediksi akhir dengan menghitung rata-rata prediksi *ntree* di hutan.

Sampel yang tidak digunakan sebagai sampel *bootstrap* pada setiap pohon dinamakan sampel *out-of-bag* (OOB). Sampel OOB dapat digunakan untuk memperkiraan kesalahan OOB (Janitza dan Hornung, 2018). Kesalahan OOB adalah rata-rata kesalahan prediksi pada setiap sampel OOB yang dihitung menggunakan prediksi dari pohon yang tidak dilatih dengan sampel OOB itu sendiri. Tingkat kesalahan OOB dapat dihitung dengan persamaan (4),

$$\text{Kesalahan OOB} = 1 - R^2_{OOB} \quad (4)$$

dengan R^2_{OOB} dapat dihitung dengan persamaan (5),

$$R^2_{OOB} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} \quad (5)$$

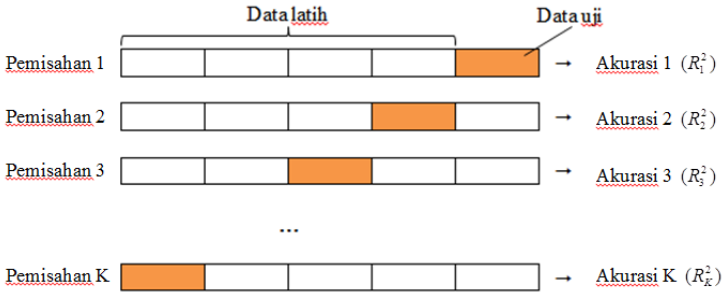
dengan y_i adalah nilai aktual sampel OOB, $\hat{y}_i$ adalah nilai prediksi sampel OOB yang diprediksi menggunakan pohon di *random forest* yang tidak berisi sampel OOB tersebut, dan $\bar{y}_i$ adalah rata-rata nilai aktual sampel OOB.

Tiga *hyperparameter* utama yang berpotensi sangat mempengaruhi kinerja model pada algoritma *random forest* yaitu *ntree*, *mtry*, dan *nodesize* (Xu *et al*., 2019). *ntree* menyatakan banyak pohon yang dibangun dalam *ensemble*. *mtry* atau ukuran set fitur adalah banyak variabel yang tersedia untuk dipecah pada setiap simpul pohon yang menentukan keragaman di antara pohon keputusan (Xu *et al*., 2019). *Hyperparameter mtry* memiliki nilai minimum 1 dan nilai maksimum sama dengan banyak variabel prediktor, *p*. *nodesize* adalah ukuran minimal sampel dari simpul daun yang digunakan sebagai kriteria penghentian dan mengontrol kedalaman pohon (Xu *et al*., 2019).

Random search CV adalah algoritma optimasi yang mengambil sampel ruang pencarian dengan keacakan atau probabilitas (Zabinsky, 2010). Algoritma ini dapat menemukan kombinasi *hyperparameter* pada model di mana kombinasi *hyperparameter*

tersebut dapat meningkatkan kinerja model (Yu dan Zhu, 2020). Beberapa tahapan dalam algoritma *random search* sebagai berikut:

1. Menentukan banyak iterasi (*n_iter*), *K*, dan ruang pencarian (*ntree, mtry, nodesize*).
2. Pilih *hyperparameter* secara acak di ruang pencarian sehingga terbentuk kombinasi *hyperparameter* sebanyak *n_iter*.
3. Ulangi langkah berikut hingga mencapai kriteria interupsi (*n_iter*)
   a. Pilih kombinasi *hyperparameter* yang sudah terbentuk.
   b. Bagi dataset menjadi *K-Folds* secara merata seperti ilustrasi pada Gambar 2.
   c. Untuk setiap *k* di mana *k* = 1, 2, 3, …, *K* dalam *K-Fold*:
      1) Melatih data latih menggunakan estimator dengan kombinasi *hyperparameter* yang sudah ditentukan.
      2) Menguji performa model pada data uji dengan skor tertentu. Pada penelitian ini menggunakan akurasi.
   d. Hitung skor akurasi rata-rata yang diperoleh..
4. Memilih kombinasi *hyperparameter* terbaik berdasarkan skor akurasi rata-rata terbesar.



Gambar 2. Ilustrasi *K-Folds Cross Validation*

Variable importance digunakan untuk melihat seberapa kuat variabel prediktor memengaruhi hasil prediksi. Algoritma untuk memperoleh *variable importance* dapat direpresentasikan seperti langkah-langkah berikut (Radohenko *et al.*, 2023):

1. Untuk setiap pohon *t* dalam *ensemble* $t \in \{1,...,T\}$:
   a. Untuk setiap simpul *i*, hitung pengurangan impuritas (seperti MSE, Gini, atau *entropy*) sebagai berikut:
   $$RI_i^{(t)} = w_i^{(t)} \cdot I_i^{(t)} - w_{LEFT_i}^{(t)} \cdot I_{LEFT_i}^{(t)} - w_{RIGHT_i}^{(t)} \cdot I_{RIGHT_i}^{(t)} \quad (6)$$
   di mana $w_i^{(t)}$, $w_{LEFT_i}^{(t)}$, dan $w_{RIGHT_i}^{(t)}$ masing-masing adalah proporsi sampel yang mencapai simpul *i* pada pohon *t*, anak simpul kiri, dan anak simpul kanan. $I_i^{(t)}$, $I_{LEFT_i}^{(t)}$, dan $I_{RIGHT_i}^{(t)}$ adalah impuritas (seperti MSE, Gini, atau *entropy*) dari simpul. Untuk simpul daun, $RI_i^{(t)} = 0$.
   b. Untuk setiap variabel *j*, hitung *variable importance* pada pohon tertentu sebagai berikut:
   $$VI_j^{(t)} = \frac{\sum_{i:\text{simpul } i \text{ tersekat variabel } j} RI_i^{(t)}}{\sum_{i \in \text{semua simpul}} RI_i^{(t)}} \quad (7)$$
2. Hitung rata-rata *variable importance* dari semua pohon dalam *ensamble* sebagai berikut:
   $$VI_j = \frac{\sum_{t=1}^{T} VI_i^{(t)}}{T} \quad (8)$$

Nilai VI yang semakin tinggi menunjukkan semakin besar kontribusi variabel prediktor tersebut mempengaruhi hasil prediksi (Argamosa *et al*., 2018).

Model yang terbentuk perlu dilakukan evaluasi kinerja model yang bertujuan untuk memperkirakan seberapa baik kinerja model yang ditentukan digeneralisasikan ke data yang tidak terlihat, yang disebut kesalahan generalisasi (Raschka, 2018). Kriteria evaluasi yang digunakan yaitu nilai *Mean Absolute Percentage Error* (MAPE). Nilai MAPE menunjukkan rata-rata kesalahan absolut dari perkiraan terhadap data aktualnya dalam bentuk persentase. MAPE dirumuskan sebagai berikut:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \qquad (9)$$

dengan n adalah banyak observasi, y_i adalah data aktual ke-i, dan $\hat{y}_i$ adalah data prediksi ke-i. Nilai MAPE dapat diinterpretasikan sesuai kriteria pada tabel berikut (Chang *et al*., 2007):

Tabel 1. Kriteria Nilai MAPE

MAPE	Interpretasi
MAPE < 10%	Kemampuan peramalan sangat baik
10% ≤ MAPE < 20%	Kemampuan peramalan baik
20% ≤ MAPE < 50%	Kemampuan peramalan cukup
MAPE ≥ 50%	Kemampuan peramalan buruk

Laptop atau terkadang disebut *notebook* adalah *Personal Computer* (PC) yang mudah dibawa dan digunakan di berbagai lokasi. Laptop dirancang untuk portabilitas sehingga sebuah laptop memiliki desain *all-in-one* yang sudah terpasang monitor, *keyboard*, *touchpad* (yang menggantikan *mouse*), dan *speaker*. Terdapat banyak pilihan produk laptop dengan spesifikasi yang berbeda-beda sehingga konsumen dapat memilih sesuai kebutuhan dan anggaran. Ukuran, kinerja yang diperlukan, sistem operasi, dan harga adalah aspek-aspek yang dapat dipertimbangkan saat mencari laptop yang tepat. Laptop dengan komponen berperforma lebih rendah memiliki harga lebih rendah, begitu juga laptop dengan komponen berperforma semakin tinggi, maka harganya semakin tinggi.

3. METODE PENELITIAN

Data yang digunakan pada penelitian ini adalah data sekunder yang diperoleh dari situs Enterkomputer, yaitu situs *ecommerce* yang menjual berbagai produk elektronik komputer degan alamat web https://www.enterkomputer.com. Pengumpulan data dilakukan menggunakan teknik *web scraping* dengan bantuan *software* ParseHub. Jumlah data yang diperoleh yaitu sebanyak 1413 produk laptop. Data tersebut berupa data produk laptop dengan variabel respon yaitu harga laptop dan variabel prediktor yaitu fitur pada laptop yang mencakup merek, prosesor, ukuran RAM, ukuran penyimpanan SSD, ukuran penyimpanan HDD, ukuran layar, dan sistem operasi.

Analisis data pada penelitian ini menggunakan bahasa pemrograman *Python* yang dijalankan melalui platform *Google Colaboratory*. Tahapan analisis data adalah sebagai berikut:

1. Mendefinisikan variabel respon dan variabel prediktor.
2. Melakukan pelabelan data pada variabel prediktor yang bersifat kategorik menggunakan *label encoder*.
3. Memisahkan data menjadi data latih dan data uji. Pada penelitian ini, data akan dibagi dengan komposisi 80% sebagai data latih dan 20% sebagai data uji.

4. Melakukan penyetelan *hyperparameter* dengan *random search CV* untuk memperoleh model regresi *random forest* yang optimal.
5. Membagun model regresi *random forest* pada data latih dengan *hyperparameter* hasil *random search CV*.
6. Mengukur kinerja model pada data latih menggunakan kesalahan OOB.
7. Melakukan prediksi pada data uji.
8. Mengevaluasi kinerja model menggunakan MAPE pada data uji.
9. Mengukur *variable importance* dari variabel prediktor.

4. HASIL DAN PEMBAHASAN

Data pada penelitian ini adalah data produk laptop dengan 8 variabel. Variabel respon yang digunakan adalah harga laptop, sedangkan variabel prediktornya adalah fitur laptop berupa merek, prosesor, ukuran RAM, ukuran penyimpanan SSD, ukuran penyimpanan HDD, ukuran layar, dan sistem operasi. Variabel bernilai kategorik pada data penelitian seperti merek, prosesor, RAM, dan sistem operasi perlu dilakukan pelabelan data. Pelabelan data dilakukan menggunakan *label encoder* dari *scikit-learn* pada *Python*. Pada dataset terdapat 12 kategori untuk variabel merek, 13 kategori untuk variabel prosesor, 6 kategori untuk variabel RAM, dan 8 kategori untuk variabel sistem operasi. Pelabelan data dimulai dari nilai 0, 1, 2, dan seterusnya secara *default* kategori diurutkan berdasarkan abjad.

Data dibagi secara acak dengan proporsi 80% sebagai data latih untuk membangun model dan 20% sebagai data uji untuk mengevaluasi kinerja model. Data keseluruhan terdapat sebanyak 1413 produk laptop yang tediri dari 1130 produk sebagai data latih dan 283 produk sebagai data uji.

Langkah selanjutnya adalah melakukan penyetelan *hyperparameter* untuk memperoleh model regresi *random forest* yang optimal. *Hyperparameter* dalam model pembelajaran mesin adalah parameter yang nilainya ditentukan sebelum memulai proses pembelajaran model. Algoritma *random search CV* untuk penyetelan *hyperparameter* yang mencakup *ntree*, *mtry*, dan *nodesize*.

Tabel 2. Ruang Pencarian *Hyperparameter*

Hyperparameter	*Random Search Value*
ntree	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
mtry	1, 2, 3, 4, 5, 6, 7
nodesize	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Tabel 2 menunjukkan ruang pencarian dari masing-masing *hyperparameter*. Proses *random search CV* pada penelitian ini melakukan pencarian acak terhadap *hyperparameter* yang diujikan dengan iterasi sebanyak 10 dan menggunakan validasi silang sebanyak 10 *fold*. Hasil dari proses *random search CV* pada iterasi 1 dengan kombinasi *hyperparameter* *ntree* = 600, *mtry* = 3, dan *nodesize* = 6 ditunjukkan pada Tabel 3.

Tabel 3. Hasil *Random Search CV* Iterasi 1

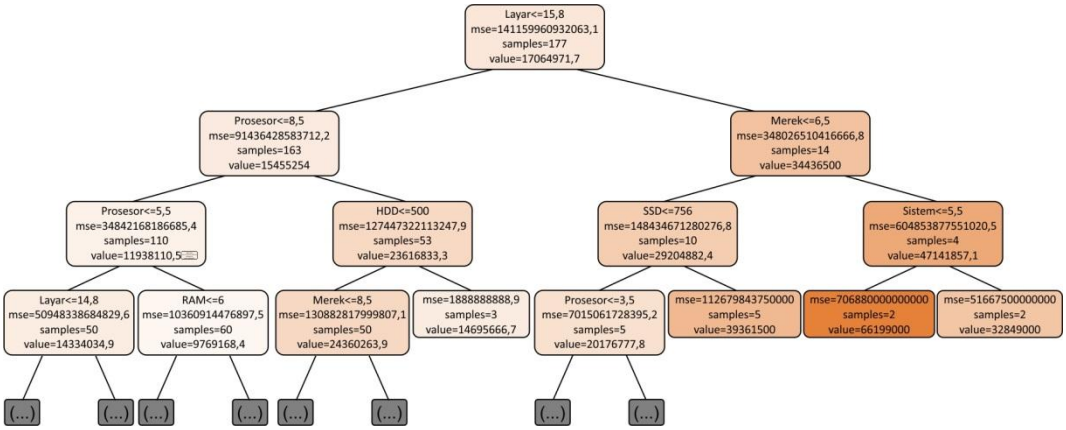
k	Skor Akurasi
1	0,907432
2	0,788339
3	0,812289
4	0,838111
5	0,895795
6	0,777486
7	0,863226
8	0,817787
9	0,863419
10	0,870616
Skor Akurasi Rata-rata	0,843450

Tabel 4 merupakan ringkasan hasil *random serach CV* dari 10 iterasi yang dijalankan. Kombinasi *hyperparameter* yang menghasilkan skor akurasi terbesar digunakan untuk membangun model regresi *random forest*. Tabel 4 menunjukkan kombinasi *hyperparameter* terbaik terdapat pada iterasi 3 dengan skor akurasi paling tinggi yaitu 0,862542. *Hyperparameter ntree* yang digunakan adalah 400, artinya terdapat 400 pohon regresi yang menyusun model regresi *random forest*. *Hyperparameter mtry* bernilai 2 artinya terdapat 2 variabel yang dipertimbangkan pada setiap simpul saat akan melakukan penyekatan. *Hyperparameter nodesize* sebanyak 2 artinya minimal terdapat 2 sampel pada setiap simpul daun.

Tabel 4. Ringkasan Hasil Penyetelan *Hyperparameter*

Iterasi	*Hyperparameter*			Skor Akurasi	Peringkat
	ntree	*mtry*	*nodesize*		
1	600	3	6	0,843450	7
2	900	1	8	0,778058	10
3	400	2	2	0,862542	1
4	900	1	2	0,852206	3
5	600	5	7	0,846370	6
6	100	5	9	0,836766	8
7	800	4	5	0,851336	4
8	800	7	10	0,836047	9
9	500	3	3	0,859143	2
10	100	6	5	0,850665	5

Tahapan selanjutnya adalah membentuk model regresi *random forest* dengan *hyperparameter* terpilih. Gambar 3 menunjukkan cuplikan dari salah satu pohon regresi dari model yang telah dibentuk. *Output* dari setiap simpul yaitu *squared error*, *samples*, dan *value*, serta aturan penyekatan pada simpul selain simpul daun. Simpul akar pada pohon regresi tersebut menyekat dengan aturan laptop berukuran layar ≤ 15,8 ke cabang kiri dan laptop berukuran layar > 15,8 ke cabang kanan di mana 15,8 merupakan nilai tengah dari nilai amatan ukuran layar. *Squared error* menunjukkan MSE yang berfungsi sebagai kriteria impuritas pada setiap penyekatan simpul. Simpul akar memiliki nilai MSE sebesar 141159960932063,1. *Samples* merupakan banyak data yang terdapat pada suatu simpul. Banyak sampel pada simpul akar yaitu 177 sampel. *Value* merupakan nilai hasil prediksi yang didapatkan dari nilai rata-rata sampel pada simpul. Nilai *value* pada simpul akar adalah 17064971,7.



Gambar 3. Pohon Regresi

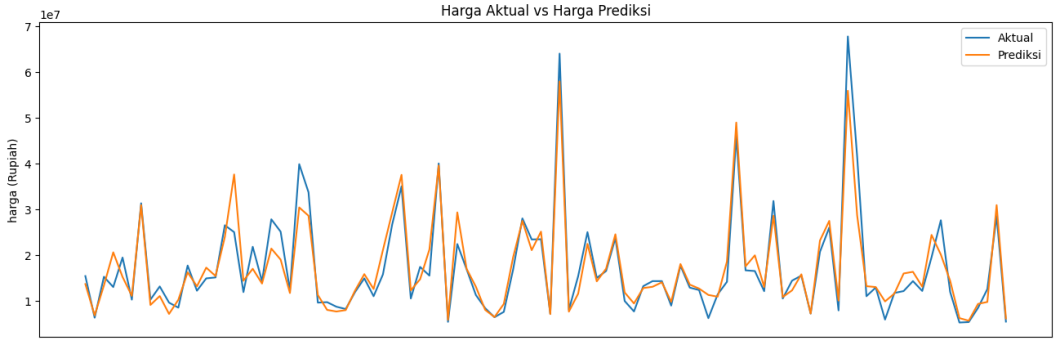
Pohon regresi pada Gambar 3 dapat ditulis ke dalam bentuk persamaan seperti berikut:

$$\hat{f}(x) = \cdots + 14695666,7 \times I\left(\text{Layar} \le 15,8\right) \times I\left(\text{Prosesor} > 8,5\right) \times I\left(\text{HDD} > 500\right) + \cdots$$
$$+ 39361500 \times I\left(\text{Layar} > 15,8\right) \times I\left(\text{Merek} \le 6,5\right) \times I\left(\text{SSD} > 756\right)$$
$$+ 66199000 \times I\left(\text{Layar} > 15,8\right) \times I\left(\text{Merek} > 6,5\right) \times I\left(\text{Sistem} \le 5,5\right)$$
$$+ 32849000 \times I\left(\text{Layar} > 15,8\right) \times I\left(\text{Merek} > 6,5\right) \times I\left(\text{Sistem} > 5,5\right)$$

di mana $I(x) = 1$ jika x adalah benar dan 0 jika x adalah salah.

Tahapan selanjutnya adalah melakukan validasi model regresi *random forest* yaitu mengukur kinerja model pada data latih menggunakan kesalahan OOB. Kesalahan OOB adalah rata-rata kesalahan prediksi pada setiap sampel *bootstrap* yang dihitung menggunakan prediksi dari pohon yang tidak dilatih dengan sampel *bootstrap* itu sendiri. Semakin rendah kesalahan OOB, maka semakin baik kinerja model tersebut. Kesalahan OOB dari model regresi *random forest* yang terbentuk sebesar 13,54% dan skor OOB sebesar 86,46% yang dapat diinterpretasikan bahwa model berkinerja baik, sehingga model dapat digunakan untuk dilanjutkan ke tahapan berikutnya.

Tahapan selanjutnya adalah melakukan evaluasi kinerja model regresi *random forest* yang terbentuk dengan membandingkan harga aktual data uji dan harga hasil prediksi data uji. Grafik perbandingan harga aktual data uji dan harga prediksi data uji ditampilkan pada Gambar 4.



Gambar 4. Grafik Harga Aktual dengan Harga Hasil Prediksi

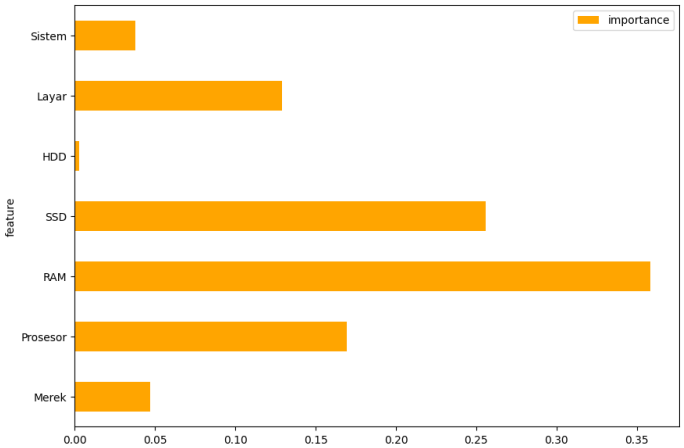
Grafik pada Gambar 4 memperlihatkan bahwa nilai prediksi yang dihasilkan oleh model memiliki nilai yang relatif mendekati nilai aktual, meskipun nilai prediksi yang dihasilkan bisa berada di atas atau di bawah dari nilai aktual. Data uji sebanyak 283 data pada Tabel 5 sebagai nilai aktual akan dibandingkan dengan nilai prediksinya, sehingga bisa dilakukan evaluasi terhadap model yang terbentuk. Tabel 5 menunjukkan bahwa nilai prediksi memiliki selisih yang relatif sedikit dari nilai aktual.

Tabel 5. Harga Aktual dan Harga Prediksi Data Uji

Data ke-i	Harga Aktual (Rp)	Harga Prediksi (Rp)
1	12999000	12271880
2	16199000	17614925
3	13799000	13659606
4	15299000	16330947
5	14899000	16330947
⋮	⋮	⋮
279	5349000	5644309
280	8449000	9320365
281	12550000	9741559
282	28499000	30899863
283	5449000	6152551

Tahapan selanjutnya adalah melakukan evaluasi kinerja model dengan menghitung MAPE pada data uji. Model regresi *random forest* yang terbentuk memiliki MAPE sebesar 14,3%. Berdasarkan kriteria penilaian MAPE pada Tabel 1, model yang terbentuk memiliki MAPE yang berada pada rentang 10% ≤ MAPE < 20% yang dapat diinterpretasikan bahwa model yang terbentuk memiliki kemampuan peramalan yang baik.

Variable importance digunakan untuk melihat seberapa kuat suatu variabel prediktor memengaruhi variabel respon. Semakin tinggi nilai VI menunjukkan semakin besar kontribusi suatu variabel pediktor tersebut memengaruhi hasil prediksi. Gambar 5 dan Tabel 6 menunjukkan *variable importance* dari masing-masing variabel prediktor. Variabel yang memiliki pengaruh paling besar terhadap prediksi harga laptop adalah RAM dengan nilai VI sebesar 0,359.



Gambar 5. Diagram *Variable Importance*

Tabel 6. Nilai *Variable Importance*

Variabel	Nilai VI
Merek	0,047
Prosesor	0,169
RAM	0,359
SSD	0,256
HDD	0,003
Layar	0,129
Sistem	0,038

Model regresi *random forest* yang terbentuk digunakan untuk melakukan percobaan prediksi harga laptop dengan kriteria laptop bermerek Acer, prosesor Intel Core i7, RAM 16 GB, SSD 512 GB, HDD 0 GB, ukuran layar 15 inci, dan sistem operasi Windows 11 Home. Hasil prediksi harga laptop tersebut sebesar Rp 14.248.788.

5. KESIMPULAN

Kesimpulan yang dapat diambil berdasarkan analisis dan pembahasan adalah bahwa metode regresi *random forest* dapat diterapkan untuk memprediksi harga laptop berdasarkan fitur laptop sehingga diperoleh pemodelan yang dapat mengetahui kisaran harga laptop berdasarkan fitur laptop yang diinginkan. Model regresi *random forest* yang terbentuk adalah regresi *random forest* dengan *hyperparameter ntree* = 400, *mtry* = 2, dan *nodesize* = 2. Model tersebut memiliki MAPE sebesar 14,3% yang menunjukkan bahwa model memiliki kemampuan peramalan yang baik. Berdasarkan nilai *variable importance*, variabel prediktor yang memberikan pengaruh paling besar terhadap prediksi harga adalah RAM dengan nilai VI = 0,359.

DAFTAR PUSTAKA

Argamosa, R. J. L. dkk., Modelling Above Ground Biomass of Mangrove Forest Using Sentinel-1 Imagery, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018, Vol. 4, No. 3: 13-20.

Breiman, L., Friedman, J., Olshen, R. A., dan Stone, C. J., *Classification and Regression Trees*, Routledge, New York, 1984.

Breiman, L., *Random* Forests, *Machine Learning*, 2001, Vol. 45, No. 1: 5-32.

Chang, P. C., Wang, Y. W., dan Liu, C. H., The Development of a Weighted Evolving Fuzzy Neural Network for PCB Sales Forecasting, *Expert Systems with Application*, 2007, Vol. 32, No. 1: 86-96.

Chen, F. H. dan Howard, H., An Alternative Model for The Analysis of Detecting Electronic Industries Earnings Management using Stepwise Regression, Random Forest, and Decision Tree, *Soft Computing*, 2015, Vol. 20, No. 5: 1945-1960.

Hastie, T., Tibshirani, R., dan Friedman, J., *The Elements of Statistical Learning: Data Mining, Interface, and Prediction (Second Edition)*, Springer, New York, 2009.

International Data Corporation (IDC), *Asia/Pacific PC Market Grew by 15.9% in 2021 Due to Continued Demand and Recovery of Supllies*, 2022, URL: https://www.idc.com/getdoc.jsp?containerId=prAP48976922.

Janitza, S. dan Hornung, R., On The Overestimation of Random Forest's Out-of-bag Error, *PLoS One*, 2018, Vol. 13, No. 8: 1-31.

Jatmiko, Y. A., Padmadisastra, S., dan Chadidjah, A., Analisis Perbandingan Kinerja CART Konvensional, Bagging dan Random Forest pada Klasifikasi Objek: Hasil dari Dua Simulasi, *Media Statistika*, 2019, Vol. 12, No. 2: 1-12.

Pal, N. dkk., How Much is My Car Worth? A Methodology for Predicting Used Cars Prices using Random Forest, *Future of Information and Communications Conference*, 2018: 1-6.

Radohenko, V., Kashnitsky, Y., dan Parshutsuch, M., *Topic 5 Ensembles and Random Forest Part 3 Feature Importance*, 2023, URL: https://mlcourse.ai/book/topic05/topic5_part3_feature_importance.html#topic05-part3.

Raschka, S., *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*, 2018, arXiv:1811.12808.

Scornet, E., Random forests and kernel methods, *IEEE Transactions on Information Theory*, 2016, Vol. 62, No. 3: 1485-1500.
Xu, Z. dkk., Water Price Prediction for Increasing Market Efficiency Using Random Forest Regression: A Case Study in the Western United States, *Water*, 2019, Vol. 11, No. 228: 1-20.
Yu, T. dan Zhu, H., Hyper-Parameter Optimization: A Review of Algorithms and Applications, 2020, arXiv abs/2003.05689.
Zabinsky, Z. B., Random Search Algorithms, *Wiley Encyclopedia of Operations Research and Management Science*, 2010: 1-13.