JUSIKOM PRIMA (Jurnal Sistem Informasi dan Ilmu Komputer Prima)
Vol. 4 No. 2, Februari 2021 E-ISSN : 2580-2879

PREDIKSI HARGA MOBIL MENGGUNAKAN ALGORITMA REGRESSI DENGAN HYPER-PARAMETER TUNING

Amalia, Muhammad Radhi, Daniel Ryan Hamonangan Sitompul, Stiven Hamonangan Sinurat, Evta Indra
Program Studi Sistem Informasi, Fakultas Teknologi dan Ilmu Komputer, Universitas Prima Indonesia
Jalan Sampul
E-mail: amalia.dhie@gmail.com

ABSTRAK- Seiring dengan bertumbuhnya tingkat aktivitas dan bisnis, mobil kini menjadi salah satu kebutuhan masyarakat. Dengan meningkatnya minat masyarakat terhadap mobil bekas, banyak orang berencana untuk memulai bisnis showroom mobil bekas. Masalah yang sering dihadapi pengusaha showroom adalah penetapan harga mobil bekas dengan tepat. Salah satu cara untuk melakukan prediksi harga adalah menggunakan metode *Machine Learning*. Untuk membuat prediksi harga yang lebih akurat dan memiliki nilai akurasi yang lebih tinggi, penelitian ini bertujuan untuk membuat sebuah model *machine learning* menggunakan algoritma regressi dengan bantuan *hyper-parameter tuning* untuk meningkatkan tingkat akurasi dari model yang dibuat dengan konfigurasi default. Model *Machine Learning* yang dibuat memiliki nilai yang berbeda, namun pada penelitian ini dipakai model *Gradient Boost Regression* yang memiliki nilai akurasi *model* sebesar 97% (setelah *tuning*) untuk melakukan prediksi. Dalam pencobaan prediksi, didapat nilai akurasi prediksi sebesar 80% dari 4 percobaan yang telah dilakukan.

Kata kunci : Machine Learning, Hyper-Parameter Tuning, Car Price Prediction, Regression Algorithm, Django

1. PENDAHULUAN

Seiring dengan bertumbuhnya tingkat aktivitas dan bisnis, mobil kini menjadi salah satu kebutuhan masyarakat. Di sisi lain, banyak fitur canggih diperkenalkan ke mobil baru, sehingga harga mobil baru meningkat signifikan. Maka dari itu, masyarakat kebanyakan memilih alternatif dengan membeli mobil bekas yang masih bagus dan layak pakai. Dengan meningkatnya minat masyarakat terhadap mobil bekas, banyak orang berencana untuk memulai bisnis showroom mobil bekas. Masalah yang sering dihadapi pengusaha showroom adalah penetapan harga mobil bekas dengan tepat. Prediksi harga mobil tetap menjadi hal yang populer [1], sehingga banyak showroom bersaing dalam harga untuk menarik pelanggan.

Salah satu cara untuk melakukan prediksi harga adalah menggunakan metode *Machine Learning*. Penelitian sebelumnya yang menggunakan metode *Machine Learning* untuk memprediksi harga kebanyakan menggunakan algoritma regressi [2-7]. Penelitian [2] menyajikan sebuah model yang dibuat dengan menggunakan algoritma *Logistic Regression* untuk memprediksi harga sembako. Model yang terdapat pada penelitian [2] tidak dibuat menggunakan *hyper-parameter tuning*, sehingga akurasi model yang didapatkan berkisar 84,2%. Penelitian [3] menyajikan sebuah model yang dibuat untuk memprediksi harga ponsel bekas dengan menggunakan algoritma *Random Forest Regressor* dan juga sama dengan penelitian [2], model yang dibuat belum menggunakan *hyper-parameter tuning*, sehingga akurasi yang didapatkan berkisar 81%.

Untuk membuat prediksi harga yang lebih akurat dan memiliki nilai akurasi yang lebih tinggi, penelitian ini bertujuan untuk membuat sebuah model *machine learning* menggunakan algoritma regressi dengan bantuan *hyper-parameter tuning* untuk meningkatkan tingkat akurasi dari model yang dibuat dengan konfigurasi default. Penelitian ini akan menyajikan tiga algoritma regressi yang sering dipakai yakni *Linear Regression, Random Forest Regression* dan *Gradient Boost Regression*. Pertama model akan dibuat berdasarkan tiga algoritma tersebut, kemudian dilakukan *hyper-parameter tuning* dan pada akhirnya akan ditentukan algoritma mana yang memiliki tingkat akurasi yang tinggi dimana model dengan tingkat akurasi tertinggi tersebut akan dipakai untuk melakukan prediksi harga mobil bekas.

2. METODOLOGI PENELITIAN

Penelitian ini dilakukan menggunakan bantuan Google Colaboratory untuk membuat model. Tahapan penelitian dapat dilihat pada Gambar 2. Penelitian ini dimulai dengan pengambilan dataset harga jual mobil bekas; melakukan pra-proses (*preprocess*) terhadap dataset yang didalamnya terdapat tahapan *Exploratory Data Analysis (EDA)* yang diakhiri dengan normalisasi data; Setelah *preprocessing* dilakukan, tiga model algoritma regressi akan dibuat yang kemudian model tersebut akan dilakukan *tuning* untuk mendapatkan nilai akurasi yang maksimal; Setelah model dibuat, model tersebut akan dipakai untuk melakukan prediksi harga mobil bekas sesuai dengan form isian yang terdapat pada halaman *web* yang dibuat menggunakan *Django* sebagai *framework*

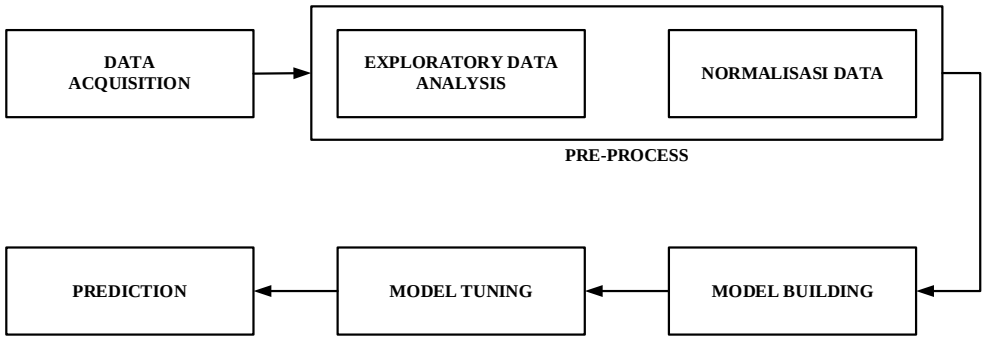
2.1 Data Acquisition

Dataset yang dipakai pada penelitian ini berisi 13 kolom dan 8128 baris. 13 kolom ini merupakan

deskripsi sebuah mobil (mulai dari nama hingga jumlah tempat duduk). Detail dataset dapat dilihat pada Gambar 2.

Gambar 1. Detail Tabel

2.2 Preprocessing



Gambar 2. Tahapan Penelitian

Pada tahap penghapusan data yang tidak memiliki nilai, digunakan fungsi drop seperti pada Gambar 3. Pada dataset yang, terdapat 5 kolom yang memiliki data-data yang memiliki nilai kosong, sehingga data yang tadinya sebanyak 8128 baris, sekarang menjadi 7906 baris setelah terkena EDA.

```
# MEMBUANG DATA-DATA YANG TIDAK BERNILAI
cars = cars.dropna(how = 'any')
cars.shape
```

Gambar 3. Fungsi Drop

Pada tahap penghapusan string untuk data yang seharusnya bernilai angka, digunakan fungsi *Regular Expression (Regex)*, sehingga kolom seperti torsi hanya terdapat nilai angka dan satuannya dibuang (*drop*). Detail tahapan ini dapat dilihat pada Tabel 1.

Tabel 1. EDA Dataset

Kolom	Sebelum EDA	Setelah EDA
Torque	190Nm@ 2000rpm	2000
Mileage	23.4 kmpl	23.4
Engine	1248 CC	1248.0
Max Power	74 bhp	74.0

2.2.2 Normalisasi Data

Tiap data dalam dataset harus di-normalisasi untuk meningkatkan *activity recognition* dengan metode *Machine Learning* [10,11]. Pada penelitian ini, normalisasi dilakukan untuk melakukan tokenization (mengubah string menjadi sebuah data angka), seperti pengubahan nilai transmisi dari Manual dan Matic menjadi nilai 1 atau 0. Fungsi yang dipakai dapat dilihat pada Gambar 4 dan detail dari *data normalization* dapat dilihat pada Tabel 2.

```
# MENGUBAH DATA KATEGORIKAL MENJADI FORMAT INTEGER

def ref1(x):
  if x == 'Manual':
    return 1
  else:
    return 0

cars_new['transmission'] = cars_new['transmission'].map(ref1)
```

Gambar 4. Tokenisasi Data

Tabel 2. Normalisasi Data

Kolom	Sebelum Normalisasi Data	Setelah Normalisasi Data
Transmisi	Manual Matic	1 0
Kategori Penjual	Individu Dealer Trustmask Dealer	1 0 -1
Bahan Bakar	Petrol Diesel LPG / CNG	1 0 -1

2.2.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) adalah proses eksplorasi data yang bertujuan untuk memahami isi dan komponen penyusun data. Tugas utama dari EDA adalah untuk membersihkan data, mendeskripsikan data, melihat persebaran data, membandingkan hubungan antar data dan menyimpulkan suatu data [8,9]. Pada penelitian ini, EDA dilakukan pada dataset untuk menghapus data-data yang tidak memiliki nilai dan menghapus string pada data yang seharusnya bernilai angka.

2.3 Model Building

Pada penelitian ini, akan dibuat tiga model regressi untuk melakukan prediksi harga, umumnya dipakai Linear Regression, Random Forest Regression dan Gradient Boost Regression.

2.3.1 Linear Regression

Algoritma *Linear Regression* ini menjelaskan relasi antara dua variabel dengan cara menyesuaikan garis regresi satu data dependent pada suatu data independent [12,13]. Pada penelitian ini, algoritma *Linear Regression* dibuat menggunakan konfigurasi *default*. Persamaan untuk pembuatan algoritma ini dapat dilihat pada (1). Pembuatan algoritma dapat dilihat pada Gambar 5.

$$Y = a * X + b \text{.....(1)}$$

Dimana Y adalah variabel tetap, a adalah slope, X adalah varibel tidak tetap dan b adalah intercept.

```
# PEMBUATAN MODEL LINEAR REGRESSION

reg = LinearRegression()
reg.fit(Xtrain, ytrain)

LinearRegression()
```

Gambar 5. Pembuatan Model Linear Regression

2.3.2 Random Forest Regression

Random Forest (RF) menghasilkan ratusan atau bahkan ribuan decision tree, yang akan menjadi fungsi regressi dan output yang dihasilkan dari RF adalah rata-rata dari tiap output dari decision tree [14,15]. Pada penelitian ini, algoritma RF dibuat menggunakan konfigurasi *n_estimators* 300; *random state* 42 dan *n_jobs* -1. Persamaan untuk pembuatan algoritma ini dapat dilihat pada (2). Pembuatan algoritma dapat dilihat pada Gambar 6.

$$n_{ij} = w_l C_j - w_{left(j)} C_{left(j)} - w_{right(j)} - C_{right(j)} \text{.....(2)}$$

n_{ij} = The importance of node j
w_l = Weighted number of samples reaching node j
C_j = The impurity value of node j
$left(j)$ = Child node from left split on node j
$right(j)$ = Child node from right split on node j [15]

```
# PERANCANGAN MODEL RANDOM FOREST
random_model = RandomForestRegressor(n_estimators=300,
                                     random_state = 42,
                                     n_jobs = -1)
```

Gambar 6. Pembuatan Model Random Forest

2.3.3 Gradient Boost Regression

Gradient Boost Regression (GBR) adalah sekumpulan decision tree untuk melakukan klasifikasi dan regresi. Cara kerja umum dari gradient boost adalah membuat urutan pohon, dimana tiap pohon berfokus pada sisa prediksi pohon-pohon sebelumnya [16,17]. Pada penelitian ini, GBR dibuat menggunakan konfigurasi *n_estimators* 400, *max_depth* 4, *min_samples_split* 2, *learning_rate* 0.1, dan memakai fungsi *loss squared_error*. Persamaan untuk pembuatan algoritma ini dapat dilihat pada (3). Pembuatan algoritma dapat dilihat pada Gambar 7.

$$\frac{1}{n}\sum_i (y_i' - y_i) \text{.....(3)}$$

y_i' = Hasil Prediksi
y_i = Nilai Observasi
n = Banyaknya sampel pada y [16]

```
# PEMBUATAN MODEL GRADIENT BOOSTING REGRESSOR

gbr = GradientBoostingRegressor(n_estimators = 400,
                                max_depth = 4,
                                min_samples_split = 2,
                                learning_rate = 0.1,
                                loss = 'ls')
```

Gambar 7. Pembuatan Model Gradient Boosting

2.4 Model Tuning

Pada penelitian ini, untuk meningkatkan akurasi dari model yang telah dibuat, maka diperlukan pemakaian *hyper-parameter* tuning. Dengan adanya tuning ini, model juga akan lebih solid/robust. Tahapannya adalah menambahkan konfigurasi baru pada model yang akan dibuat, lalu seluruh model akan ditrain ulang. Parameter yang ditambahkan pada *Linear Regression* adalah intercept, untuk *Random Forest* ditambahkan *max features* 'sqrt' dan *criterion 'gini'*, sementara untuk gradient boost ditambahkan *max features 'sqrt'* dan *criterions 'friedman_mse'*. Proses tuning dapat dilihat pada Gambar 8.

```
# TUNING

cv = 10

model_to_test = ['LinearRegression', 'RandomForest', 'GradientBoostingRegressor']

regression_dict = dict(LinearRegression = linear_model,
                       RandomForest = forest_model,
                       GradientBoostingRegressor = gbr_model)

param_grid_dict = dict(LinearRegression = param_grid_linear,
                       RandomForest = param_grid_forest,
                       GradientBoostingRegressor = param_grid_gbr)
```

Gambar 8. Pembuatan Tuning Pada Model

3. KESIMPULAN

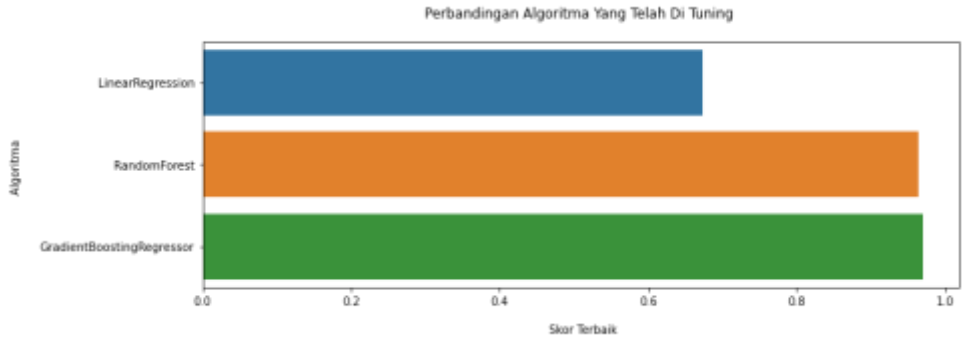
3.1 Hasil Training Model

Hasil dari training model yang didapat dari ketiga model adalah *Linear Regression* mendapat akurasi *training* sebesar 69% dan akurasi *testing* sebesar 67,28%; *Random Forest* mendapat akurasi

training sebesar 99,17% dan akurasi *testing* 96,3%; dan *Gradient Boost* mendapat akurasi *training* 99,58% dan akurasi *testing* 96,75%. Dari hasil ini, disimpulkan bahwa algoritma yang mendapatkan nilai akurasi model *testing* terbesar adalah *Gradient Boost*, sementara *Linear Regression* mendapat nilai akurasi model *testing* terkecil dan juga *overfitting*.

3.2 Hasil Tuning Model

Hasil dari tuning model uang didapat dari ketiga model adalah Linear Regression tidak mendapat perubahan akurasi, Random Forest mendapat perubahan sebesar 0,02% menjadi 96,32% dari nilai akurasi sebelumnya dan Gradient Boost mendapat perubahan sebesar 0,25% menjadi 97% dari nilai akurasi sebelumnya. Komparsi ketiga model ini akan disajikan dalam bentuk diagram blok sebagaimana pada Gambar 9.



Gambar 9. Komparasi Algoritma Regressi Setelah *Hyper-Tuning*

Gambar 10. Prediksi Pada Web Berbasis Framework Django

3.3 Prediksi

Prediksi akan dilakukan pada website berbasis framework dari Django. User hanya perlu melakukan input data sesuai form yang tersedia kemudian melakukan proses yang akhirnya model akan menampilkan berapa prediksi harga mobil. Beberapa pencobaan prediksi telah peneliti lakukan dan harga prediksi dibandingkan dengan range harga mobil bekas yang dipasarkan di website e-commerce OLX. Detail dari prediksi dijabarkan pada Tabel 3. Form pada web dapat dilihat pada Gambar 10.

Tabel 3. Pencobaan Prediksi

Mobil	**Harga Prediksi**	**Range Harga E-Commerce**	**T/F**
Avanza 2019	Rp 179.176.468,12	Rp 160-195 Juta	T
Pajero 2018	Rp 376.894.802,70	Rp 439-460 Juta	F
Escudo	Rp. 65.842.211,17	Rp 63 – 69 Juta	T
Innova 2020	Rp 402.156.440,18	Rp 395-410 Juta	T
Mobil	**Harga Prediksi**	**Range Harga E-Commerce**	**T/F**
Ertiga 2017	Rp 155.924.502,29	Rp 149-160 Juta	T
Jumlah Prediksi Benar			4
Akurasi			80%

4. PENUTUP

Pengaruh teknologi yang mulai menyebar secara masif ke seluruh bidang memang dapat membantu

siapapun yang memerlukannya. Prediksi harga mobil bekas yang telah dibuat menggunakan metode *Machine Learning* ini juga dapat membantu pengusaha *showroom* untuk menjual mobil dengan harga yang wajar sesuai dengan range harga yang sering ditemui. Model *Machine Learning* yang dibuat memiliki nilai yang berbeda, namun pada penelitian ini dipakai model *Gradient Boost Regression* yang memiliki nilai akurasi *model* sebesar 97% (setelah *tuning*) untuk melakukan prediksi. Dalam pencobaan prediksi, didapat nilai akurasi prediksi sebesar 80% dari 4 percobaan yang telah dilakukan.

DAFTAR PUSTAKA

[1] A. Pandey, V. Rastogi, and S. Singh, "Car's selling price prediction using random forest machine learning algorithm," SSRN Electronic Journal, 2020.

[2] V. W. Siburian and I. E. Mulyana, "Prediksi Harga Ponsel Menggunakan Metode Random Forest," Prosiding Annual Research Seminar 2018, vol. 4, no. 1, pp. 144–147, 2018.

[3] A. Saiful, "Prediksi Harga Rumah Menggunakan web scrapping Dan Machine learning Dengan Algoritma linear regression," JATISI (Jurnal Teknik Informatika dan Sistem Informasi), vol. 8, no. 1, pp. 41–50, 2021.

[4] T. D. Phan, "Housing price prediction using machine learning algorithms: The case of Melbourne City, Australia," 2018 International Conference on Machine Learning and Data Engineering (iCMLDE), 2018.

[5] N. Pal, P. Arora, P. Kohli, D. Sundararaman, and S. S. Palakurthy, "How much is my car worth? A methodology for predicting used cars' prices using random forest," Advances in Intelligent Systems and Computing, pp. 413–422, 2018.

[6] C. Ma, Z. Liu, Z. Cao, W. Song, J. Zhang, and W. Zeng, "Cost-sensitive deep forest for price prediction," Pattern Recognition, vol. 107, p. 107499, 2020.

[7] N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya, and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), 2018.

[8] A. Kumar, Learning predictive analytics with python: Gain practical insights into predictive modelling by implementing predictive analytics algorithms on public datasets with pyton. Packt Publishing, 2016.

[9] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and Deep Learning," Future Generation Computer Systems, vol. 81, pp. 307–313, 2018.

[10] I. M. Pires, F. Hussain, N. M. M. Garcia, P. Lameski, and E. Zdravevski, "Homogeneous data normalization and Deep Learning: A Case Study in human activity classification," Future Internet, vol. 12, no. 11, p. 194, 2020.

[11] H.-C. Huang and L.-X. Qin, "Empirical evaluation of data normalization methods for molecular classification," PeerJ, vol. 6, 2018.

[12] B. Sravani and M. M. Bala, "Prediction of student performance using linear regression," 2020 International Conference for Emerging Technology (INCET), 2020.

[13] S. Liu, M. Lu, H. Li, and Y. Zuo, "Prediction of gene expression patterns with generalized linear regression model," Frontiers in Genetics, vol. 10, 2019.

[14] Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J. C.-W. Chan, P. van den Bossche, J. Van Mierlo, and N. Omar, "Random Forest regression for online capacity estimation of lithium-ion batteries," Applied Energy, vol. 232, pp. 197–210, 2018.

[15] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, Random Forest and KNN models for the text classification," Augmented Human Research, vol. 5, no. 1, 2020.

[16] S. H. Samadi, B. Ghobadian, and M. Nosrati, "Prediction of higher heating value of biomass materials based on proximate analysis using gradient boosted regression trees method," Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 43, no. 6, pp. 672–681, 2019.

[17] F.-K. Wang and T. Mamo, "Gradient boosted regression model for the degradation analysis of prismatic cells," Computers & Industrial Engineering, vol. 144, p. 106494, 2020.

[18] K. Shankar, Y. Zhang, Y. Liu, L. Wu, and C.-H. Chen, "Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification," IEEE Access, vol. 8, pp. 118164–118173, 2020.

[19] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using Spatial Data," Ecological Modelling, vol. 406, pp. 109–120, 2019.