Jurnal Elektronik Ilmu Komputer Udayana
Volume 12, No 1. August 2023

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Pendekatan *Deep Learning* dan *Gradient Boosting* dalam Prediksi Harga Properti *Airbnb* dengan Analisis Sentimen

Christopher Digno[a1], Muhammad Iqbal Jauhar[a2], Muhammad Nur Syaifullah[a3]

[a]Program Studi Informatika, Fakultas Teknologi Informasi dan Sains Data,
Universitas Sebelas Maret
Jebres, Surakarta, Indonesia

[1]digno.christopher@student.uns.ac.id (Corresponding author)
[2]iqbaljauhar@student.uns.ac.id
[2]muhammad.nur.syaiful@student.uns.ac.id

Abstrak

Penentuan harga properti sewa Airbnb yang sesuai untuk mendapatkan penjualan yang tertinggi merupakan pekerjaan yang tidak mudah, terlebih pada masa modern sekarang yang dipenuhi dengan pasar bebas dan pertarungan harga yang seringnya tidak sehat. Dalam waktu yang sama, calon penyewa properti juga kesulitan melakukan penilaian atas harga yang ditawarkan oleh pemilik properti. Oleh karena itu, kami menawarkan beberapa model machine learning untuk melakukan prediksi harga Airbnb. Kami berhasil mendapatkan hasil terbaik menggunakan XGBoost dengan MSE (Mean Squared Error) sebesar 0.1414. Selanjutnya, kami juga melakukan pembenahan terhadap metode seleksi fitur yang digunakan pada penelitian sebelumnya dengan menggunakan ElasticNet dan berhasil menurunkan MSE dari 0.1471 menjadi 0.1370.

Kata Kunci: *Machine Learning, Deep Neural Network, XGBoost, Price Prediction, Regression*

1. Pendahuluan

Pemilik properti pada platform makelar properti *Airbnb* sering kali menemui kesulitan dalam menentukan harga sewa dari properti mereka yang masuk pada *listing* karena pemilihan harga sewa memiliki dampak yang sangat besar pada jumlah tamu yang akan menyewa properti tersebut. Fakta bahwa *Airbnb* memiliki dampak yang besar bagi perekonomian kota metropolitan [1], [2] membuat pemilihan harga sewa menjadi hal yang vital bagi perekonomian daerah tersebut. Di sisi lain, tamu yang ingin menyewa properti di *Airbnb* mungkin tidak bisa melakukan evaluasi harga properti yang optimal karena data yang terbatas.

Penelitian ini berupaya untuk membantu menyelesaikan permasalahan mengenai kesulitan penentuan dan evaluasi harga di atas. Kami akan membangun sebuah model prediksi harga menggunakan konsep-konsep *machine learning* (ML)*, deep learning* (DL)*,* dan *natural language processing* (NLP) yang dapat digunakan untuk pemilik properti serta tamu yang tertarik untuk menyewa properti. Data yang akan digunakan untuk membangun model kami adalah data mengenai fitur-fitur properti Airbnb dan daftar ulasan/*review* dari properti tersebut. Kami akan mengulas beberapa metode yang sudah pernah digunakan sebelumnya, seperti regresi linear, model berbasis *tree*, *Support Vector Regression* (SVR), *K-means Clustering*, dan *Neural Networks* (NN), dan juga mengajukan beberapa metode baru seperti *Deep Neural Network* (DNN), *Extreme Gradient Bossting* (*XGBoost*), *Bayesian Regression*, dan *K-Nearest Neighbors Regressor*. Selain melakukan komparasi terhadap beberapa metode tersebut, kami juga akan membandingkan performa dua metode seleksi fitur, yaitu *Lasso* (berbasis L1) dan *ElasticNet* (kombinasi antara penalti L1 dan L2).

Terdapat beberapa penelitian terdahulu yang mencoba untuk membangun model prediksi harga baik untuk properti rental berbasis *non-sharing* atau personal. Penelitian oleh Yu dan Wu [3] mengajukan model prediksi harga perumahan menggunakan regresi linear, SVR, dan *Random Forest regressor* ditambah dengan analisis fitur, dan menghasilkan RMSE (*root mean squared error*) sebesar 0.53, serta implementasi PCA (*Principle Component Analysis*) dengan SVC (*State Vector Classifier*) yang

menghasilkan akurasi sebesar 69%. Penelitian lain [4] mendapatkan model terbaiknya menggunakan *Regression Tree* dengan RMSE sebesar 1.05 CNY/m^2-*day*.

Di luar banyak penelitian di atas mengenai prediksi properti rental berbasis *non-sharing* dan hotel/sejenisnya, terdapat beberapa penelitian untuk properti rental berbasis *sharing*, yang lebih berhubungan dengan *branding Airbnb*. Penelitian oleh Kalehbasti, et al. [5] menggunakan beberapa metode regresi untuk memprediksi harga properti *Airbnb* menggunakan L1 *regularization*, dengan SVR sebagai model terbaik mereka yang menghasilkan MSE sebesar 0.6692. Beberapa metode regresi lain juga digunakan pada beberapa penelitian lainnya, seperti OLS (*Ordinary Least Squares*) [6], [7] dan *Quantile Regression* [7]. Penelitian oleh Yang et al. [8] menawarkan metode Regresi Linear yang jauh lebih sederhana, menggunakan penilaian pengguna pada properti sewa. Selain metode berbasis linear di atas, Li et al. [9] memperkenalkan metode *clustering* berbasis *Multi-scale Affinity*.

Penelitian ini berupaya untuk menghadirkan metode regresi yang baru dengan mengimplementasikan *Deep Neural Networks, XGBoost*, dan beberapa model linear lainnya. Kami juga membandingkan dua jenis metode seleksi fitur, *Lasso* dan *ElasticNet*, menggunakan SVR sebagai metode regresi yang saya gunakan sebagai *benchmark*.

2. Metode Penelitian

Pada bagian ini, kami akan mengulas langkah-langkah yang kami lakukan dalam penelitian ini. Pertama, kami akan membahas *dataset* yang akan kami gunakan. Kemudian, kami akan menjelaskan bagaimana kami akan melakukan *preprocessing* pada *dataset* tersebut. Selanjutnya kami akan menjelaskan dua eksperimen yang akan kami lakukan pada penelitian ini, yaitu pembuatan dan penyetelan (*tuning*) model *machine learning* dan studi komparatif dua jenis seleksi fitur berbasis *Lasso* dan *ElasticNet*.

2.1. *Dataset*

Dalam penelitian ini, kami akan menggunakan sebuah *dataset* yang disediakan oleh *Inside Airbnb* (http://insideairbnb.com/). *Inside Airbnb* merupakan sebuah proyek komunitas yang memiliki tujuan untuk menunjukkan dampak dari *Airbnb* untuk lingkungan suatu kota/daerah. Data yang kami gunakan adalah detail *listing* properti pada *website* dan kumpulan ulasan dari *listing* properti tersebut di kota New York City, New York, Amerika Serikat [10]. Gambar 1 dan Tabel 1 memberikan visualisasi dan beberapa jenis statistik mengenai *dataset*.

Gambar 1. Distribusi geografis dari *dataset listing* kota New York City. Gambar merupakan tangkapan layar dari http://insideairbnb.com/new-york-city/

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Tabel 1. Jumlah baris dan fitur (kolom) untuk masing-masing data ulasan dan *listing*.

	Data ulasan	**Data *listing***
Jumlah baris	1.051.974	50.220
Jumlah fitur	6	96

2.2. *Preprocessing* Data

Setelah data yang dibutuhkan selesai diunduh, kami kemudian melakukan serangkaian langkah *preprocessing*. Langkah-langkah ini dilakukan supaya *dataset* yang kami miliki dapat digunakan untuk melatih model *machine learning* kami. Untuk *preprocessing* awal, kami melakukan beberapa rekayasa pada *dataset listing* kami, antara lain:

a. Menghapus fitur-fitur yang tidak penting/tidak memiliki korelasi terhadap harga, misalnya nama *host* dan deskripsi properti.
b. Melakukan *one hot encoding* pada beberapa fitur yang berupa data kategori, kualitatif, dan *array*.
c. Membersihkan simbol-simbol pada data, seperti simbol mata uang dolar ($).
d. Membersihkan nilai-nilai kosong dengan menjadikannya sebagai sebuah nilai tertentu atau membuat baris tersebut.
e. Mengubah data tanggal menjadi jumlah hari menuju tanggal tertentu.

Setelah melakukan beberapa langkah *preprocessing* di atas, dimensi data kami meledak dengan cukup signifikan pada sumbu kolom karena banyak fitur yang dijadikan *one hot encoding*. Tabel 2 menunjukkan perbandingan statistik data *listing*.

Tabel 2. Perbandingan jumlah baris dan fitur (kolom) data *listing* sebelum dan sesudah *preprocessing.*

	Data *listing* sebelum	**Data *listing* sesudah**
Jumlah baris	50.220	49.984
Jumlah fitur	96	764

2.2.1. Analisis Sentimen Data Ulasan

Supaya data ulasan properti dapat kami gunakan untuk melakukan prediksi harga properti, kami harus melakukan analisis sentimen dari ulasan-ulasan tersebut. Analisis sentimen merupakan sebuah teknik analisis opini yang sering digunakan pada banyak penelitian untuk mengetahui opini seseorang akan suatu hal [11]. Pada umumnya, model analisis sentimen menerima masukan sebuah kalimat dan mengeluarkan sebuah nilai dengan rentang -1 hingga 1 yang berturut-turun dapat diartikan sebagai sentimen yang sangat negatif dan sentimen yang sangat positif.

Pada data ulasan penelitian ini, kami melakukan analisis sentimen untuk setiap item ulasan menggunakan pustaka *TextBlob* [12]. Selanjutnya, kami melakukan pengelompokan untuk ulasan pada satu properti yang sama dengan nilai rata-ratanya, serta menghilangkan fitur lain selain *listing ID* dan nilai sentimen. Karena pengelompokan ini, jumlah data ulasan kami menyusut.

Tabel 3. Perbandingan jumlah baris dan fitur data ulasan sebelum dan setelah analisis sentimen

	Data ulasan sebelum	**Data ulasan sesudah**
Jumlah baris	1.051.974	39.528
Jumlah fitur	6	2

2.2.2. Seleksi Fitur

Dataset listing dan ulasan yang sudah digabung memiliki jumlah fitur yang sangat besar, yaitu 765. Dari banyak fitur tersebut, tidak semuanya memiliki kontribusi yang besar dalam melakukan prediksi terhadap harga properti. Terlebih lagi, menggunakan data dengan fitur yang terlalu banyak dapat menyebabkan model *machine learning* mendapatkan pengukuran performa yang buruk karena gagal untuk melakukan generalisasi atas data yang ada [13].

Metode seleksi fitur yang akan kami gunakan pada *preprocessing* adalah *Lasso* (*Least Absolute Shrinkage and Selection Operator*). *Lasso* merupakan sebuah metode analisis regresi yang dapat digunakan untuk seleksi fitur dan regularisasi. *Lasso* menggunakan norma L1 untuk melakukan regularisasi, berdampak ke hasil koefisien yang lebih jarang (*sparse*), dan terkadang menghasilkan nilai koefisien 0 [14]. Persamaan objektif yang dijadikan sebagai tujuan optimasi *Lasso* dapat dilihat pada (1) [14].

$$\min_{\beta \in \mathbb{R}^p} \{ \frac{1}{N} \left| |y - X\beta| \right|_2^2 \} \quad (1)$$

Untuk melakukan seleksi fitur *Lasso* pada data, kami akan menggunakan pustaka *scikit-learn* yang mendukung *Lasso regularizer* dan *feature selection*. Selain itu, untuk memastikan performa model linear yang dibangun, kami akan melakukan *hyperparameter tuning* menggunakan banyak nilai *alpha*. Setelah menjalankan model dan mendapatkan hasil terbaik, kami menggunakan model tersebut untuk menentukan nilai koefisien *Lasso* yang kemudian digunakan untuk menentukan apakah fitur tersebut sesuai untuk digunakan atau tidak. Pada akhirnya, kami membuang fitur data yang memiliki koefisien sebesar 0. Setelah dilakukan seleksi fitur, jumlah fitur dalam data ditunjukkan pada Tabel 4.

Tabel 4. Perbandingan jumlah baris dan fitur data sebelum dan sesudah seleksi fitur

	Data sebelum	**Data sesudah**
Jumlah baris	49.976	49.976
Jumlah fitur	764	197

Selanjutnya pada tahap eksperimen 2, kami akan membandingkan hasil seleksi fitur ini dengan metode seleksi fitur yang menggabungkan norma L1 dan L2, yaitu *ElasticNet*.

2.2.3. Pemisahan dan Normalisasi Data

Seperti metode *machine learning* lainnya, data yang digunakan akan dibelah menjadi 3 bagian, yaitu data *training*, *validation*, dan *testing*. Kami menggunakan 10% data untuk *testing*, 10% untuk *validation* dan 90% untuk *training*. Meskipun proporsinya yang cukup jauh, karena *dataset* yang kami gunakan memiliki data berjumlah besar, strategi pemisahan kami dapat tetap dipakai.

Setelah data selesai dipisah, kami melakukan normalisasi data. Normalisasi berarti mengubah semua nilai yang ada dalam sebuah data sehingga semua nilai tersebut memiliki skala yang sama. Dalam *machine learning*, data yang memiliki beragam jenis skala akan menghasilkan model yang memiliki bias tinggi ke fitur yang memiliki data berskala besar [15]. Kami akan melakukan normalisasi data menggunakan strategi normalisasi *Min-max*, yang mengubah sebuah nilai dalam sebuah fitur berdasarkan nilai maksimal dan minimalnya.

2.3. Pembuatan dan Penyetelan Model *Machine Learning*

Kami membagi eksperimen pertama kami menjadi dua tahap, yaitu eksperimen menggunakan *Deep Neural Network* dan kumpulan metode regresi tradisional beserta *XGBoost.*

2.3.1. Pembuatan dan Penyetelan Model *Deep Neural Network*

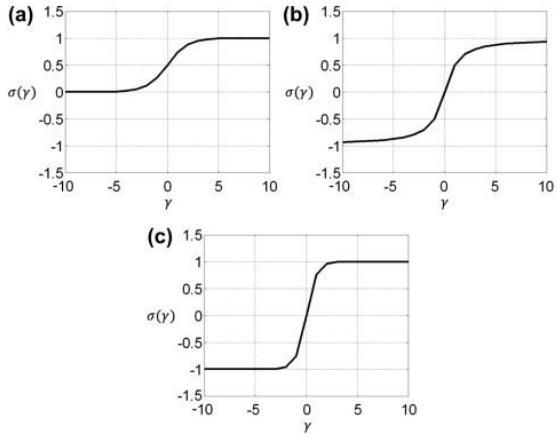
Deep Neural Network (DNN) adalah salah satu metode klasifikasi dan regresi yang sering digunakan pada domain *machine learning*. Algoritma DNN mirip dengan *Artificial Neural Network* (ANN) yang sama-sama berusaha untuk meniru cara otak manusia melakukan pemrosesan informasi [16]. Perbedaan fundamental DNN dengan *Neural Network* (NN) adalah kompleksitas arsitekturnya. DNN memiliki lebih dari satu *hidden layer* yang ada ditengah-tengah *input* dan *output layer*. Masing-masing *layer* memiliki sekumpulan *neuron*, yang terhubung dengan *neuron* lainnya dengan konfigurasi tertentu, seperti struktur jaringan saraf yang ada pada manusia. Namun pada implementasinya, banyak penelitian yang menggunakan istilah NN untuk merujuk pada arsitektur *Neural Network* dengan lebih dari satu *hidden layer* [17].

Pada masing-masing *node* dalam masing-masing *layer*, sebuah fungsi transformasi dikenakan pada *input*, yang disebut *activation function* [17]. *Activation function* bertujuan untuk memungkinkan DNN

p-ISSN: 2301-5373
e-ISSN: 2654-5101

menyelesaikan permasalahan non-linear. Ada banyak jenis *activation function* yang dapat digunakan sesuai dengan kebutuhan, namun secara fundamental, *activation function* melakukan transformasi terhadap *input* sesuai dengan sebuah fungsi linear yang ditetapkan [18].

Pada penelitian ini, kami akan melakukan implementasi DNN dengan banyak *hidden layer* sejumlah dua, tiga, empat, dan lima. Beberapa pilihan *hidden layer* ini dibuat untuk mempelajari efek jumlah *hidden layer* terhadap kualitas generalisasi yang dilakukan, serta meneliti permasalahan *vanishing gradient*. Kami akan melakukan *hyperparameter tuning* menggunakan *Keras Tuner* yang disediakan oleh pustaka DNN yang kami gunakan, *Keras*, dengan strategi *Hyperband tuning*, sebuah pendekatan *tuner* berbasis bandit yang meraih performa *state of the art* dibanding banyak strategi *hyperparameter tuning* lainnya [19].



Gambar 2. Perbandingan grafik fungsi beberapa *activation function*, yaitu (a) *sigmoid*, (b) *arc-tangent*, dan (c) *hyperbolic tangent*. Gambar diambil dari Bakr and Negm [20].

2.3.2. Pembuatan dan Penyetelan Model Regresi Tradisional

Sebagai pembanding untuk metode DNN, kami akan membangun beberapa model regresi berbasis *machine learning* tradisional. Yang dimaksud dengan *machine learning* tradisional adalah model-model *machine learning* yang tidak menggunakan konsep-konsep *deep learning*, membutuhkan banyak *feature engineering* dalam proses *learning* [21].

Ada beberapa model regresi tradisional yang akan kami implementasi pada eksperimen ini, yaitu *Bayesian Regression*, *K-Nearest Neighbors Regressor*, dan *XGBoost.*

a. *Bayesian Regression*

Bayesian Regression adalah sebuah model regresi yang didasari dengan teori *Bayes*, yang mendeskripsikan probabilitas sebuah kejadian, dengan diketahui hal-hal yang mungkin berhubungan dengan kejadian tersebut. *Bayesian Regression* mengadopsi pendekatan probabilitas untuk menentukan nilai-nilai parameter, berkebalikan dengan OLS (*Ordinary Least Squares*) yang mengasumsikan parameter yang tetap, cenderung mengabaikan ketidakpastian *estimator*.

Pada penelitian ini, kami menggunakan varian *Bayesian Ridge Regression*, di mana *ridge penalty* diberlakukan supaya koefisien dapat diperkecil, mengurangi permasalahan *overfitting*.

b. *K-Nearest Neighbors Regressor*

K-Nearest Neighbors (K-NN) *regressor* adalah sebuah model regresi yang melakukan *training* dan prediksi atas nilai target/label berdasarkan rata-rata dari nilai target dari beberapa nilai terdekatnya pada ruang fitur [22]. K-NN tidak memiliki asumsi terkait korelasi/keterkaitan data secara fungsional, namun model regresi K-NN melakukan prediksi berdasarkan pola lokal dari data yang ada.

c. *XGBoost*

XGBoost (*Extreme Gradient Boosting*) adalah sebuah model *machine learning* berbasis *gradient boosting decision tree*. Secara prinsip, *XGBoost* menggunakan metode *gradient descent* untuk membuat banyak *decision tree* baru berdasarkan *tree* yang ada sebelumnya dengan harapan untuk melakukan minimalisasi fungsi objektif [23]. Dalam fungsi objektif *XGBoost*, terdapat persamaan mengenai *loss function* dan *regularization term*, yang tentunya berhubungan dengan koefisien penalti. Fungsi objektif ini dijelaskan pada (2) [23].

$$Obj(\theta) = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda x} + \lambda T \quad (2)$$

Pada penelitian ini, kami menggunakan dua metode pengukuran performa (*performance metrics*) untuk mengukur kualitas dari model *machine learning* kami, yaitu MSE dan R^2. Kami menggunakan pustaka bawaan *TensorFlow* untuk melakukan pengukuran performa.

a. MSE (*Mean Squared Error*)

MSE melakukan penilaian berdasarkan rata-rata dari kuadrat eror, yaitu jarak antara persamaan garis prediksi dan nilai yang sesungguhnya, untuk masing-masing titik observasi. Persamaan dari MSE dinotasikan dengan (3).

$$MSE(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad (3)$$

b. R^2 (*Coefficient of Determination*)

Sedikit berbeda dari MSE, R^2 merupakan pengukuran proporsi dari varians variabel dependen yang dapat diprediksi dari variabel independen. Secara sederhana R^2 merupakan MSE yang sudah dinormalisasi dengan skala dari data, sehingga lebih mudah diinterpretasikan. Persamaan R^2 dinotasikan dengan (4).

$$R^2\left(Y, \hat{Y}\right) = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \hat{Y}\right)^2} \quad (4)$$

Untuk masing-masing model yang akan kami bangun di atas, sama halnya dengan model DNN, kami akan melakukan *hyperparameter tuning* untuk memastikan bahwa kami dapat meraih hasil terbaik dari masing-masing model yang kami gunakan. Untuk model *Bayesian Regression* dan *K-Nearest Neighbors Regressor*, kami akan menggunakan strategi *brute force* untuk *hyperparameter tuning*, dengan algoritma sederhana yang kami tulis sendiri. Sedangkan untuk model *XGBoost*, kami akan menggunakan pustaka *Hyperopt* dengan algoritma *Bayesian Optimization* yang menggunakan *Bayesian inference* dan model probabilitas untuk menemukan satu set *hyperparameter* yang optimal dengan diberikan sebuah ruang pencarian.

2.4. Studi Komparatif Seleksi Fitur

Dalam bagian sebelumnya mengenai pembangunan model *deep neural network* dan *machine learning* tradisional, kami menggunakan seleksi fitur berbasis *Lasso*. Selain *Lasso*, ada berbagai macam jenis seleksi fitur lainnya yang dapat kami bandingkan performanya dengan *Lasso*. Untuk itu, kami memilih *ElasticNet* sebagai objek studi komparatif kami, karena merupakan pengembangan dari algoritma *Lasso* dengan penambahan metode regularisasi L2, yang sering disebut *Ridge*. Kombinasi dari dua metode regularisasi ini bertujuan untuk menghindari kekurangan dari masing-masing metode regularisasi dan menggabungkan kelebihannya. Jika norma L1 melakukan penyusutan koefisien bahkan sampai 0 [14], norma L2 lebih fokus pada melakukan penyusutan koefisien yang lebih kecil namun tidak sampai 0, mengurangi kuantitas fitur *colinear*.

Sebagai perbandingan dari kedua metode seleksi fitur ini, kami akan melakukan *fitting* model SVR untuk *dataset* yang sudah diseleksi fitur menggunakan *ElasticNet* dan membandingkannya dengan *dataset* yang diseleksi menggunakan *Lasso* dari penelitian Kalehbasti et al [5]. Indikator keberhasilan komparasi ini adalah nilai MSE yang paling rendah (R^2 yang paling tinggi).

p-ISSN: 2301-5373
e-ISSN: 2654-5101

3. Hasil dan Pembahasan

Pada bagian ini, kami akan menunjukkan hasil dari beberapa eksperimen kami. Kami juga akan mencoba untuk menjelaskan alasan dari hasil tersebut. Dalam melakukan eksperimen 1 dan 2, kami menggunakan *kernel Jupyter Notebook* dari *Kaggle* varian gratis (tidak berbayar), yang menawarkan layanan *cloud notebook* menggunakan CPU secara gratis. Secara kumulatif dari kedua eksperimen, *kernel* kami menghabiskan waktu kira-kira 18 jam.

3.1. Eksperimen 1 (Pembuatan dan Penyetelan Model *Machine Learning*)

Dalam proses pemisahan data (*dataset train/val/test split*), kami menggunakan 10% data / 4.998 baris untuk masing-masing *test* dan *validation set* dan sisanya, 80% data / 39.980 baris, kami gunakan untuk *training set*. Kami menggunakan model terbaik dari penelitian Kalehbasti et al. [5], yaitu SVR dengan nilai performa yang ditunjukkan pada **Tabel 5**.

Tabel 5. Model *baseline* yang akan kami gunakan sebagai perbandingan. Tabel diadaptasi dari penelitian Kalehbasti et al. [5]

Model	MSE	R^2
SVR (*baseline*)	**0.1471**	0.6901

Tabel 6 menunjukkan model terbaik untuk masing-masing eksperimen DNN dengan ukuran *layer* (2, 3, 4, dan 5 *layer*) beserta konfigurasinya dibandingkan dengan model *baseline*. Nilai *performa* untuk masing-masing model dilihat dari *best epoch* untuk masing-masing sesi *training*, bukan pada *epoch* terakhir.

Catatan: Kolom konfigurasi berarti jumlah *dense layer* untuk masing-masing *layer* dipisahkan dengan *dash* (-). Contoh: 250 – 250 – 250 berarti terdapat *input layer*, 3 *dense layer*, dan *output layer* (sejumlah 1 untuk semua model). Kemudian, entri *dropout* berarti terdapat *dropout* layer sebesar 0.2 pada lokasi tersebut.

Tabel 6. Perbandingan model terbaik dari eksperimen DNN

Model	Konfigurasi	Pengukuran Perfoma	
		MSE	R^2
DNN 2-*layer*	365 – 365 – *dropout*	0.1558	0.6715
DNN 3-*layer*	455 – 125 – 35	0.1564	0.6703
DNN 4-*layer*	485 – 485 – 305 – 335	0.1541	0.6753
Baseline SVR		**0.1471**	**0.6901**

Tabel 7 menunjukkan model terbaik untuk masing-masing hasil eksperimen metode *machine learning* tradisional.

Tabel 7. Perbandingan model terbaik dari eksperimen *machine learning* tradisional

Model	Pengukuran Perfoma	
	MSE	R^2
Bayesian Regression	0.1553	0.6741
K-NN *Regression*	0.2112	0.5567
XGBoost	**0.1414**	**0.7031**
Baseline SVR	0.1471	0.6901

Dari hasil eksperimen DNN di atas, dapat dilihat bahwa tidak ada model DNN yang dapat mengungguli hasil dari model *baseline* SVR. Dari semua model DNN, model 4-*layer* menghasilkan performa yang terbaik, meskipun dengan margin yang sangat sedikit dari model-model lainnya. Dalam permasalahan ini, menambah *layer* dari DNN tidak membantu untuk menambah performanya.

Hal ini disebabkan oleh DNN dengan *layer* besar sangat rawan terkena *overfitting*, karena banyaknya *trainable parameters* yang ada di dalam arsitekturnya. Kemungkinan permasalahan kedua adalah *learning rate* yang terlalu besar, sehingga menyebabkan *oscillating learning*. Hal ini sudah kami coba mitigasi dengan menggunakan *learning rate* yang sangat kecil, yaitu $1 \cdot 10^{-4}$, namun tidak menghasilkan kenaikan yang signifikan.

Selanjutnya, untuk eksperimen *machine learning* tradisional, model *XGBoost* yang kami buat dapat mengungguli hasil *baseline* dengan nilai MSE sebesar 0.1414, yang bukan merupakan merupakan kenaikan yang signifikan. *XGBoost* sebagai satu-satunya model dari eksperimen ini yang berhasil meraih performa di atas *baseline* menunjukkan kekuatan *XGBoost* sebagai model *ensemble* yang terdiri atas banyak *tree*, yang dapat bekerja dengan data yang tidak *linearly separable*.

Dari dua bagian dari eksperimen pertama di atas, dapat ditunjukkan bahwa *XGBoost* sebagai model yang secara relatif lebih sederhana daripada SVM yang lebih rumit dan DNN yang jauh lebih rumit dapat mendapatkan hasil yang terbaik.

3.2. Eksperimen 2 (Studi Komparatif Seleksi Fitur)

Kami menggunakan skema pemisahan data yang sama dengan eksperimen sebelumnya dan mengubah metode seleksi fitur. Tabel 8 merupakan perbandingan antara dua metode seleksi fitur dengan metode *machine learning* yang sama, yaitu SVM.

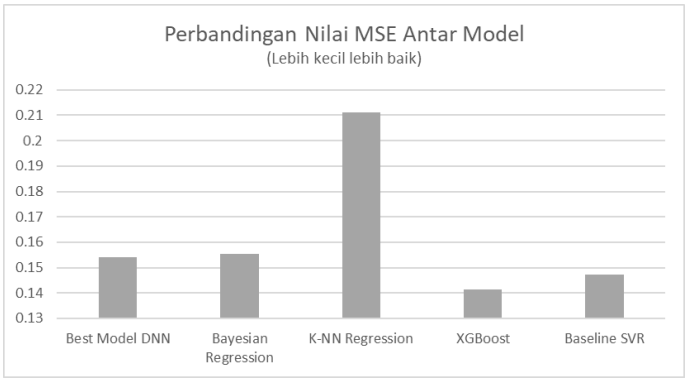
Tabel 8. Perbandingan antara metode seleksi fitur *ElasticNet* dan *Lasso*

Metode Seleksi Fitur	Pengukuran Perfoma	
	MSE	R^2
ElasticNet	**0.1370**	**0.7101**
Baseline Lasso	0.1471	0.6901

Model seleksi fitur *ElasticNet* menunjukkan kemampuan yang lebih tinggi untuk mengidentifikasi fitur yang lebih penting untuk digunakan pada metode *machine learning*. Seperti yang telah diulas sebelumnya pada bagian metode, *ElasticNet* merupakan penyempurnaan dari model seleksi fitur berbasis norma penalti L1 (*Lasso*) dan L2 (*Ridge*), menghasilkan fitur yang diseleksi menggunakan penalti L1 dan L2 yang imbang.

4. Kesimpulan

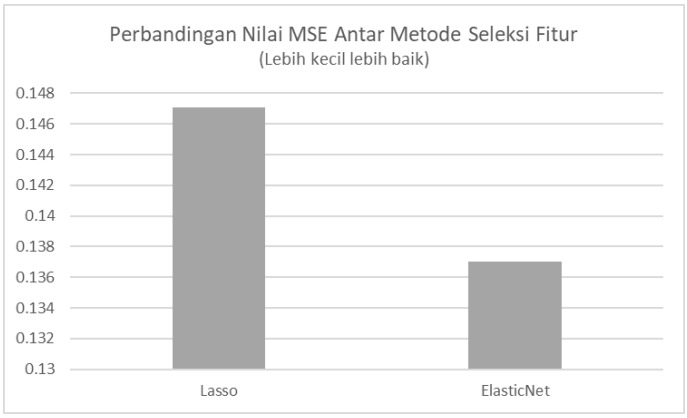
Penelitian ini memiliki tujuan untuk memperbaiki hasil penelitian mengenai deteksi harga *Airbnb* dari penelitian-penelitian yang ada sebelumnya. Kebaruan yang ditawarkan dari penelitian ini adalah model-model *machine learning* baru, termasuk *Deep Neural Network*, *Bayesian Regression*, K-NN *Regression*, dan *XGBoost*. Metode-metode baru tersebut kami *train* menggunakan *dataset* dan *feature engineering* yang sama dengan penelitian sebelumnya, dan kami dapat memberikan hasil yang lebih baik. Dengan pengukuran performa berbasis MSE (*Mean Squared Error*) dan R^2, kami berhasil membuat model *XGBoost* dengan nilai MSE sebesar 0.1414 dan R^2 sebesar 0.7031.



Gambar 3. Perbandingan nilai MSE antar model *machine learning* yang dibuat

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Kami juga melakukan eksperimen menggunakan metode seleksi fitur baru, yaitu *ElasticNet*, yang merupakan penyempurnaan dari metode seleksi yang dipakai sebelumnya, *Lasso*. Kami berhasil mendapatkan hasil yang lebih baik dengan metode *machine learning* SVR sebagai *benchmark*, dengan MSE sebesar 0.1370 dan R^2 sebesar 0.7101.



Gambar 4. Perbandingan nilai MSE antar metode seleksi fitur yang dibandingkan

Penelitian yang kami buat pada eksperimen pertama menggunakan seleksi fitur *LASSO* yang berfokus pada *baseline* dibandingkan dengan model baru sedangkan untuk eksperimen kedua kita menggunakan *ElasticNet* yang berfokus pada perbandingan antara *LASSO* dengan *ElasticNet.* Jadi antara eksperimen pertama dan kedua memiliki konteks yang berbeda.
Ada beberapa penelitian lanjutan yang dapat kami usulkan, yaitu (1) pembuatan paradigma *feature engineering* yang sepenuhnya berbeda dengan metode seleksi fitur yang berbeda juga seperti *Principle Component Analysis* (PCA), (2) penelitian mengenai masalah-masalah klasik pada *Deep Neural Network* seperti *vanishing gradient boosting* pada permasalahan ini, dan (3) menggunakan *dataset* yang berbeda sebagai objek penelitian, seperti hotel atau layanan akomodasi berbasis pariwisata lainnya yang mungkin memiliki permasalahan penentuan harga yang sama dengan penelitian kami.

Referensi

[1] F. Tian, F. Sun, B. Hu, and Z. Dong, “The Impact on Bed and Breakfast Prices: Evidence from Airbnb in China,” *Sustainability*, vol. 14, no. 21, Art. no. 21, Jan. 2022, doi: 10.3390/su142113834.
[2] G. Zervas, D. Proserpio, and J. W. Byers, “The Rise of the Sharing Economy: Estimating the Impact of Airbnb on the Hotel Industry,” *J. Mark. Res.*, vol. 54, no. 5, pp. 687–705, Oct. 2017, doi: 10.1509/jmr.15.0204.
[3] H. Yu and J. Wu, “Real Estate Price Prediction with Regression and Classification CS 229 Autumn 2016 Project Final Report,”
[4] Y. Ma, Z. Zhang, A. Ihler, and B. Pan, “Estimating Warehouse Rental Price using Machine Learning Techniques,” *Int. J. Comput. Commun. Control*, vol. 13, no. 2, pp. 235–250, Apr. 2018, doi: 10.15837/ijccc.2018.2.3034.
[5] P. R. Kalehbasti, L. Nikolenko, and H. Rezaei, “Airbnb Price Prediction Using Machine Learning and Sentiment Analysis,” 2021, pp. 173–184. doi: 10.1007/978-3-030-84060-0_11.
[6] D. Wang and J. L. Nicolau, “Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on Airbnb.com,” *Int. J. Hosp. Manag.*, vol. 62, pp. 120–131, Apr. 2017, doi: 10.1016/j.ijhm.2016.12.007.
[7] L. Masiero, J. L. Nicolau, and R. Law, “A demand-driven analysis of tourist accommodation price: A quantile regression of room bookings,” *Int. J. Hosp. Manag.*, vol. 50, pp. 1–8, Sep. 2015, doi: 10.1016/j.ijhm.2015.06.009.

[8] Y. Yang, N. J. Mueller, and R. R. Croes, “Market accessibility and hotel prices in the Caribbean: The moderating effect of quality-signaling factors,” *Tour. Manag.*, vol. 56, pp. 40–51, Oct. 2016, doi: 10.1016/j.tourman.2016.03.021.
[9] Y. Li, Q. Pan, T. Yang, and L. Guo, “Reasonable price recommendation on Airbnb using Multi-Scale clustering,” *2016 35th Chin. Control Conf. CCC*, pp. 7038–7041, Jul. 2016, doi: 10.1109/ChiCC.2016.7554467.
[10] “Airbnb Public Dataset.” http://insideairbnb.com/get-the-data/ (accessed Jun. 13, 2023).
[11] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artif. Intell. Rev.*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.
[12] S. Loria, “Textblob: simplified text processing,” 2018.
[13] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, “Selecting critical features for data classification based on machine learning methods,” *J. Big Data*, vol. 7, no. 1, p. 52, Jul. 2020, doi: 10.1186/s40537-020-00327-4.
[14] S. Zhang, F. Zhu, Q. Yu, and X. Zhu, “Identifying DNA-binding proteins based on multi-features and LASSO feature selection,” *Biopolymers*, vol. 112, no. 2, p. e23419, 2021, doi: 10.1002/bip.23419.
[15] M. Kang and J. Tian, “Machine Learning: Data Pre-processing,” in *Prognostics and Health Management of Electronics*, John Wiley & Sons, Ltd, 2018, pp. 111–130. doi: 10.1002/9781119515326.ch5.
[16] A. R. N. Aouichaoui, R. Al, J. Abildskov, and G. Sin, “Comparison of Group-Contribution and Machine Learning-based Property Prediction Models with Uncertainty Quantification,” in *Computer Aided Chemical Engineering*, M. Türkay and R. Gani, Eds., in 31 European Symposium on Computer Aided Process Engineering, vol. 50. Elsevier, 2021, pp. 755–760. doi: 10.1016/B978-0-323-88506-5.50118-2.
[17] I. G. N. A. Indrawan and I. M. Widiartha, “Optimization Artificial Neural Network Using Artificial Bee Colony in Letter Recognition Classification,” *JELIKU J. Elektron. Ilmu Komput. Udayana*, vol. 8, no. 4, pp. 469–473, 2020, doi: 10.24843/JLK.2020.v08.i04.p13.
[18] J. Lim *et al.*, “Development of Dye Exhaustion Behavior Prediction Model using Deep Neural Network,” in *Computer Aided Chemical Engineering*, Y. Yamashita and M. Kano, Eds., in 14 International Symposium on Process Systems Engineering, vol. 49. Elsevier, 2022, pp. 1825–1830. doi: 10.1016/B978-0-323-85159-6.50304-3.
[19] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization,” *J. Mach. Learn. Res.*, vol. 18, no. 185, pp. 1–52, 2018.
[20] M. H. Bakr and M. H. Negm, “Chapter Three - Modeling and Design of High-Frequency Structures Using Artificial Neural Networks and Space Mapping,” in *Advances in Imaging and Electron Physics*, M. J. Deen, Ed., in Silicon-Based Millimeter-wave Technology, vol. 174. Elsevier, 2012, pp. 223–260. doi: 10.1016/B978-0-12-394298-2.00003-X.
[21] Y. Kumar *et al.*, “Heart Failure Detection Using Quantum-Enhanced Machine Learning and Traditional Machine Learning Techniques for Internet of Artificially Intelligent Medical Things,” *Wirel. Commun. Mob. Comput.*, vol. 2021, p. e1616725, Dec. 2021, doi: 10.1155/2021/1616725.
[22] Murni, R. Kosasih, A. Fahrurozi, T. Handhika, I. Sari, and D. P. Lestari, “Travel Time Estimation for Destination In Bali Using kNN-Regression Method with Tensorflow,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 854, no. 1, p. 012061, May 2020, doi: 10.1088/1757-899X/854/1/012061.
[23] S. Pan, Z. Zheng, Z. Guo, and H. Luo, “An optimized XGBoost method for predicting reservoir porosity using petrophysical logs,” *J. Pet. Sci. Eng.*, vol. 208, p. 109520, Jan. 2022, doi: 10.1016/j.petrol.2021.109520.