Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 4, August 2024, pp. 1149-1164

DOI: https://doi.org/10.52436/1.jutif.2024.5.4.2565
p-ISSN: 2723-3863
e-ISSN: 2723-3871

OPTIMAL STUDY OF REAL-ESTATE PRICE PREDICTION MODELS USING MACHINE LEARNING

Ikhsan Maulana[*1], Amril Mutoi Siregar[2], Santi Arum Puspita Lestari[3], Sutan Faisal[4]

[1,2,3,4]Informatics Engineering Department, Faculty of Computer Science, Universitas Buana Perjuangan Karawang, Indonesia
Email: [1]if20.ikhsanmaulana@mhs.ubpkarawang.ac.id, [2]amril.mutoi@ubpkarawang.ac.id, [3]santiarum@ubpkarawang.ac.id, [4]sutanfaisal@ubpkarawang.ac.id

(Article received: July 23, 2024; Revision: August 14, 2024; published: September 3, 2024)

Abstract

Everyone wants a place to live, especially close to work, shopping centers, easy transportation, low crime rates and others. Pricing must also pay attention to external factors, not just the house. Determining this price is sometimes difficult for some people. Therefore, the aim of this research is to predict real-estate prices by taking these factors into account. Prediction results are very useful for sellers who have difficulty determining prices and also for prospective buyers who are confused when making financial plans to buy a house in the desired neighborhood. The dataset used in this research was obtained from Kaggle and consists of 506 samples with 14 attributes. Several machine learning algorithms, such as Extra Trees (ET), Support Vector Regression (SVR), Random Forest (RF), eXtreme Gradient Boosting (XGB), Gradient Boosting Machine (GBM), Light Gradient Boosting Machine (LGBM), and CatBoost, used to predict real-estate prices. This research uses Principal Component Analysis (PCA) for feature selection techniques in data sets after the preprocessing phase and before model building. The highest accuracy model obtained is CatBoost with GridSearchCV, this model has been cross validated so there is very little chance of overfitting when given new data. The SVR model with a poly kernel uses a Principal Component (PC) of 10 and GridSearchCV gets an R2 Score of 0,87, a very large number close to the score of CatBoost with GridSearchCV.

Keywords: *boosting, deep learning, machine learning, optimized model.*

STUDI OPTIMAL MODEL PREDIKSI HARGA REAL-ESTATE MENGGUNAKAN MACHINE LEARNING

Abstrak

Semua orang menginginkan tempat tinggal terutama dekat dengan tempat kerja, pusat perbelanjaan, kemudahan transportasi, tingkat kriminalitas yang rendah dan lainnya. Penentuan harga juga harus memperhatikan faktor eksternal tersebut tidak hanya rumahnya saja. Penentuan harga ini terkadang sulit bagi sebagian orang. Oleh karena itu, tujuan penelitian ini adalah untuk memprediksi harga *real-estate* dengan memperhatikan faktor tersebut. Hasil prediksi sangat berguna bagi penjual yang kesulitan menentukan harga dan juga bagi calon pembeli yang kebingungan ketika membuat rencana keuangan untuk membeli rumah di lingkungan yang diinginkan. Dataset yang digunakan dalam penelitian ini diperoleh dari Kaggle dan terdiri dari 506 sampel dengan 14 atribut. Beberapa algoritma pembelajaran mesin, seperti *Extra Trees* (ET), *Support Vector Regression* (SVR), *Random Forest* (RF), *eXtreme Gradient Boosting* (XGB), *Gradient Boosting Machine* (GBM), *Light Gradient Boosting Machine* (LGBM), dan *CatBoost*, digunakan untuk memprediksi harga *real-estate*. Penelitian ini menggunakan *Principal Component Analysis* (PCA) untuk teknik pemilihan fitur pada kumpulan data setelah fase prapemrosesan dan sebelum pembuatan model. Model akurasi tertinggi yang didapatkan adalah *CatBoost* dengan *GridSearchCV*, model ini telah dilakukan *Cross Validation* sehingga kemungkinan sangat kecil terjadi *overfitting* ketika diberi data baru. Model SVR dengan *kernel poly* menggunakan *Principal Component* (PC) sebanyak 10 dan *GridSearchCV* mendapat R^2 *Score* sebesar 0,87, angka yang sangat besar mendekati skor yang dimiliki oleh *CatBoost* dengan *GridSearchCV*.

Kata kunci: *boosting, deep learning, machine learning, model optimal.*

1. PENDAHULUAN

Pasar Real Estate merupakan salah satu bagian paling penting dalam perekonomian sebuah negara

[1]. Seiring waktu, semakin banyak orang yang membutuhkan perumahan, terutama di dekat tempat kerja, pusat perkantoran, pusat perbelanjaan, kemudahan transportasi dan lainnya. Tentu saja hal tersebut akan dengan cepat mempengaruhi harga real estate. Umumnya masyarakat menggunakan *House Price Index* (HPI) [2]. HPI merupakan indikasi estimasi harga yang diperoleh dengan mengukur rata-rata perubahan harga pada penjualan berulang [2]. HPI tidak efisien dalam memprediksi harga rumah tertentu. Daripada hanya mengandalkan harga jual selama beberapa dekade sebelumnya, terdapat banyak faktor yang perlu dipertimbangkan, seperti fasilitas lingkungan sekitar, usia bangunan, jumlah lantai dan lainnya [3]. Model penetapan harga hedonis juga digunakan untuk memperkirakan harga rumah dengan menentukan faktor eksternal (kode pos, fasilitas umum dan lainnya.) dan internal (kamar tidur, kamar mandi, dapur, ruang keluarga dan seterusnya) [4]. Penjual yang hanya menggunakan metode tradisional mungkin akan merugi karena harga yang ditetapkan tanpa mempertimbangkan banyak faktor, sehingga bisa saja harga terlalu murah ataupun terlalu mahal. Oleh karena itu, dibutuhkan solusi yang efektif dan komprehensif untuk memprediksi harga real estate. Model dan algoritma *machine learning* telah terbukti menjadi solusi yang efektif dan komprehensif untuk berbagai masalah dalam memprediksi harga real estate, harga saham dan lainnya [5], [6].

Machine learning kini menjadi pendekatan prediksi yang paling efektif karena dapat memprediksi lebih akurat berdasarkan beberapa faktor, terlepas dari data penjualan tahun sebelumnya. Sejumlah penelitian sebelumnya oleh [7] menunjukkan algoritma *Linear Regression*, *Polynomial Regression*, *Decision Tree* dan *Random Forest* mendapatkan nilai *Mean Absolute Percentage Error* (MAPE) masing - masing sebesar 34,42%, 30,15%, 25,42%, 20,56%, 21,81%. Penelitian lain oleh [8] menggunakan algoritma *Lasso Regression* dengan nilai *alpha* 2 dan *selection random* mendapatkan akurasi sebesar 72%. Penelitian [9] menggunakan *deep learning* yaitu ElasticNet dan ANN mendapatkan skor MAE sebesar 62,921, 56,128, angka yang cukup besar yang mengindikasikan bahwa model tersebut kurang baik dalam melakukan tugasnya. Penelitian yang dilakukan oleh [10] menggunakan model *deep learning* yang dioptimalkan *hyperparameter* memberikan skor R^2 yang cukup mengesankan, ANN tanpa optimasi mendapat skor R^2 sebesar 0.90, sedangkan ANN dengan optimasi sebesar 0.95. *Hyperparameter* yang digunakan yaitu ReLU sebagai fungsi aktivasi, Adam sebagai algoritma optimasi, *batch size* 550, *dropout* 0.005, *learning rate* 0.0012 dan *validation split* sebesar 8%. Selain optimasi *hyperparameter*, penggabungan ANN dengan algoritma lain dapat memberikan akurasi yang lebih baik, seperti penelitian [11]. Penelitian tersebut menggabungkan ANN dengan algoritma *Levenberg-Marquart*, *Bayesian Regularization* dan *Scaled Conjugate Gradient*. ANN dengan algoritma *Levenberg-Marquart* yang memiliki 15-20 *neuron* di *hidden layer* mendapat skor R^2 sebesar 0,9436, ANN dengan algoritma *Bayesian Regularization* memiliki 10-20 *neuron* di *hidden layer* mendapat skor R^2 sebesar 0,9749, dan terakhir ANN dengan algoritma *Scaled Conjugate Gradient* yang memiliki 20 *neuron* di *hidden layer* mendapat skor R^2 sebesar 0,9315. [12] meneliti algoritma berbasis boosting seperti GBR, XGBM dan LGBM untuk memprediksi harga rumah menunjukkan kinerja yang baik dan pengaruh *overfitting* lebih kecil dari algoritma *Random Forest* dan *Extra Trees*. Algoritma *Extra Trees* mendapatkan skor R^2 sebesar 0,9995, akan tetapi algoritma tersebut mengalami *overfitting* jika dilakukan *cross validation*. Algoritma *boosting* (GBR, XGBM, LGBM) *overfitting* hanya sebesar 7.9% - 8.3% saat *cross validation* walaupun skor R^2 tidak sebesar algoritma *Random Forest* dan *Extra Trees*. Algoritma *Random Forest* dan *Extra Trees* dapat bekerja dengan baik, namun mengalami *overfitting* lebih tinggi, sehingga dapat mengurangi generalisasi terhadap data baru. *Overfitting* disebabkan oleh meningkatnya kompleksitas model, yang mungkin diakibatkan oleh peningkatan varians prediksi [12].

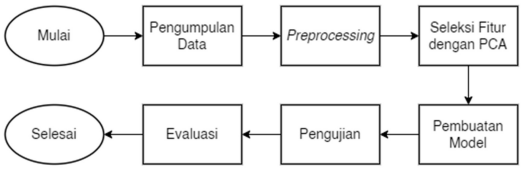
Metodologi pada penelitian ini membutuhkan penggunaan algoritma regresi *supervised learning*. Penelitian ini menggunakan 7 algoritma regresi, yaitu *Extra Trees* (ET), *Support Vector Regression* (SVR), *Random Forest* (RF), *eXtreme Gradient Boosting* (XGB), *Gradient Boosting Machine* (GBM), *Light Gradient Boosting Machine* (LGBM), dan *CatBoost*. Teknik PCA juga digunakan untuk menyeleksi fitur serta *GridSearchCV* untuk optimasi *hyperparameter*. Multikolinearitas yang tinggi dapat menjadi masalah karena meningkatkan varians estimasi koefisien dan membuat estimasi menjadi sangat sensitif terhadap perubahan kecil dalam model [13]. Ketidakstabilan estimasi koefisien menyulitkan interpretasi model. *Principal Component Analysis* (PCA) adalah metode *unsupervised machine learning* yang digunakan untuk mengurangi dimensi tinggi dalam kumpulan data [14]. Konsep PCA memproyeksikan kumpulan data berdimensi tinggi ke dalam sub-ruang yang lebih rendah, dimana komponen utama diidentifikasi dari sekumpulan fitur yang tidak berkorelasi [15]. Oleh karena itu, PCA mengekstrak fitur utama sambil mempertahankan varians penting dalam atribut fitur asli. Hal ini memungkinkan untuk mengidentifikasi dan memberi peringkat pada fitur berpengaruh yang meningkatkan akurasi prediksi.

Berdasarkan permasalahan yang telah diuraikan sebelumnya dan dengan dukungan hasil penelitian yang telah dilakukan sebelumnya, penelitian ini melakukan komparasi berbagai algoritma regresi *supervised learning*. Penggunaan

teknik PCA juga diimplementasikan untuk menyeleksi fitur serta mengoptimalkan *hyperparameter* menggunakan GridSearchCV, agar mendapatkan kombinasi *hyperparameter* yang terbaik. Setiap hasil regresi dievaluasi dengan berbagai metrik, seperti R^2 *Score*, MAE, MSE, dan RMSE. Hasil prediksi dengan algoritma *supervised learning* dapat digunakan sebagai acuan para penjual real estate, agar harga dapat disesuaikan dengan berbagai faktor sehingga tidak hanya mengandalkan data penjualan tahun sebelumnya. Hasil prediksi juga dapat digunakan oleh pembeli untuk membantu menentukan apakah harga rumah dibawah harga, diatas harga atau harga sesuai/normal.

2. METODE PENELITIAN

Tahapan yang dilakukan dalam penelitian ini diawali dengan tinjauan literatur. Penelusuran literatur dilakukan dengan tujuan untuk mengidentifikasi landasan teori yang digunakan dan mencari literatur ilmiah yang relevan untuk mendukung penelitian. Tahapan keseluruhan penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Diagram alir tahapan penelitian

2.1. Pengumpulan data

Dataset Real-Estate diperoleh dari situs *Kaggle* oleh Arslan Ali, yang bekerja sebagai *Data Engineer* di perusahaan Techlogix Lahore, Punjab, Pakistan. Dataset dapat diakses di URL berikut: https://www.kaggle.com/datasets/arslanali4343/real-estate-dataset. Kumpulan data ini berdasarkan pada informasi yang dikumpulkan oleh U.S. *Census Service* mengenai perumahan di Boston, Massachusetts. Kumpulan data berisi 506 sampel dan 14 Atribut. Penjelasan setiap variabel disajikan pada Tabel 1.

Tabel 1. Deskripsi variabel dataset

Tipe	Atribut
CRIM	Tingkat kejahatan per kota.
ZN	Proporsi lahan perumahan yang dikategorikan untuk lahan seluas lebih dari 25.000 kaki persegi.
INDUS	Proporsi luas usaha non-ritel per kota dalam satuan hektar.
CHAS	Variabel sungai Charles (1 jika bangunan dekat sungai, 0 jika tidak)
NOX	Konsentrasi oksida nitrat (bagian per 10 juta).
RM	Jumlah rata-rata kamar per hunian.
AGE	Proporsi unit yang ditempati pemilik yang dibangun sebelum tahun 1940.
DIS	Jarak tertimbang ke lima pusat ketenagakerjaan Boston.
RAD	Indeks aksesibilitas terhadap jalan raya radial.
TAX	Tarif pajak properti nilai penuh per $10.000.
PTRATIO	Rasio murid - guru menurut kota.
B	1000 * (Bk-0.63)2 dimana Bk adalah proporsi penduduk kulit hitam menurut kota.
LSTAT	Persentase status yang lebih rendah dari populasi.
MEDV	Nilai median rumah yang ditempati pemilik adalah $1000-an.

2.2. *Pre-processing*

Preprocessing merupakan tahapan untuk menghilangkan permasalahan yang dapat mempengaruhi hasil pengolahan data [16]. Fase prapemrosesan melibatkan penggunaan berbagai teknik untuk menghasilkan data berkualitas tinggi. Proses yang digunakan yaitu:

1. Proses pertama yaitu pembersihan data. Pembersihan data adalah proses memperbaiki atau menghapus data yang salah, formatnya salah, duplikat, atau tidak lengkap dalam kumpulan data [17]. Dataset menjalani proses pembersihan menyeluruh untuk memastikan kualitas yang optimal. Gambar 2 terlihat bahwa kolom RM memiliki 5 data yang kosong, lalu dilakukan *preprocessing* sehingga seperti Gambar 3.

```
Total Number of Missing Value
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         5
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

Gambar 2. Atribut sebelum pre-processing

```
Total Number of Missing Value
After Preprocessing
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

Gambar 3. Atribut sesudah pre-processing

2. Selanjutnya dilakukan juga penskalaan dan normalisasi data dengan tujuan normalisasi data menggunakan penskalaan Z-score dan mencapai standarisasi. Rumus yang digunakan dalam proses normalisasi ini adalah sebagai berikut (1).

$$Z = \frac{X-\mu}{\sigma} \quad (1)$$

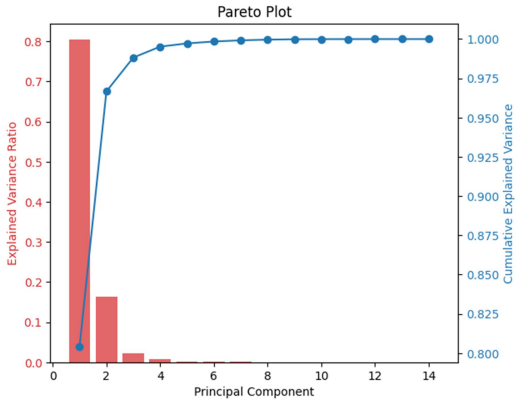
X, standarisasi fungsi merupakan aspek penting yang perlu diperhatikan saat menganalisis data. Simbol μ mewakili nilai rata-rata seluruh kumpulan data, dan σ mewakili nilai deviasi standar.

Dataset terdiri dari 506 sampel data yang dibagi menjadi data latih dan data uji. Rasio kedua kumpulan data diatur ke 80:20. Hal itu berarti data pelatihan menyumbang 80% dan data pengujian menyumbang 20%. Data latih berjumlah 405 sampel dan data uji berjumlah 101 sampel. Langkah pertama setelah membagi data adalah melakukan teknik PCA untuk menyeleksi fitur pada dataset. Data tersebut kemudian dimasukkan ke dalam model dan dilakukan proses pengujian.

2.3. Seleksi fitur dengan PCA

Fase seleksi fitur dalam *machine learning* adalah proses memilih fitur paling relevan yang tersedia dalam kumpulan data untuk dimasukkan ke model. Pemilihan fitur biasanya mempercepat pelatihan dan mempengaruhi nilai akurasi secara langsung. Teknik PCA dapat mengurangi dimensi dataset dan meningkatkan generalisasi model sekaligus meminimalkan hilangnya data sampel [18]. Setiap komponen utama merupakan kombinasi linier dari variabel asli dan tidak berhubungan, variabel asli dapat diwakili oleh data berdimensi lebih rendah dengan asumsi kehilangan informasi lebih sedikit. Dataset yang digunakan dalam penelitian ini memiliki 14 fitur. Fitur yang terlalu banyak dan korelasi antar data dapat menghalangi model untuk mencapai hasil terbaik dan menyebabkan *overfitting*. Oleh karena itu, penelitian ini menggunakan PCA pada dataset untuk mengubah 13 fitur menjadi 6 fitur dan meningkatkan performa hasilnya.

Penelitian ini menggunakan diagram *pareto*, seperti yang ditunjukkan pada Gambar 4. Diagram *pareto* untuk memeriksa nilai *eigen* dan menentukan jumlah *Principal Components* (PC) yang optimal untuk proses pemodelan selanjutnya [19]. Pada diagram *pareto* kurva PCA *pareto*, sumbu x biasanya mewakili bilangan komponen, sedangkan sumbu y mewakili nilai tunggal atau varians yang dijelaskan oleh setiap komponen. Diagram *pareto* menunjukkan bahwa terdapat 6 PC yang memiliki kontribusi daripada PC lainnya, dengan *cumulative explained variance* sebesar 0,9 . Konsep *cumulative explained variance* merupakan aspek mendasar dari PCA, yang merupakan teknik reduksi dimensi dalam analisis data multivariat. Selanjutnya, menggunakan fitur tersebut dalam data pelatihan dan pengujian untuk digunakan lebih lanjut dalam pemodelan.



Gambar 4. Diagram pareto

2.4. Ikhtisar Model

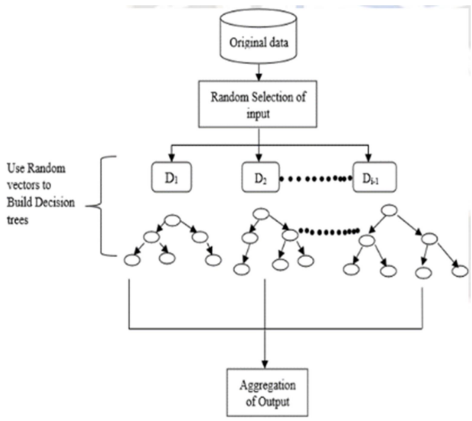
1). *Extra Trees* (ET)

Extra Trees Regression, atau sering disebut sebagai *Extremely Randomized Trees Regression*, adalah algoritma *machine learning* yang termasuk dalam keluarga algoritma pohon keputusan. Algoritma ini mirip dengan *Random Forest*, namun memiliki variasi tambahan dalam proses pemilihan fitur saat membangun pohon. ET membuat proses pemilihan fitur menjadi lebih acak dan agresif. Setiap pemisahan pada sebuah node di pohon dilakukan dengan memilih nilai acak dari seluruh ruang fitur, bukan memilih nilai terbaik. Pendekatan ini bertujuan untuk meningkatkan variasi antar pohon dan mengurangi *overfitting*. Prediksi dilakukan dalam klasifikasi dengan pemungutan suara terbanyak dari pohon keputusan. Langkah dalam ET yaitu [20] :

1. Pemilihan input secara acak.

2. Gunakan vektor acak untuk membangun beberapa pohon keputusan.
3. Gabungkan pohon keputusan.
4. Prediksi hasilnya.

Gambar 5 menunjukkan algoritma ET yang membangun kumpulan pohon keputusan dan menggabungkan prediksinya untuk membuat prediksi akhir.



Gambar 5. Proses *extra tree*. Sumber: Kundra et al [20]

2). *Support Vector Regression* (SVR)

Support Vector Regression (SVR) adalah algoritma *machine learning* yang digunakan untuk memodelkan dan memprediksi hubungan antara variabel dependen dan independen. Tujuan utama SVR adalah menemukan fungsi yang dapat memberikan prediksi yang konsisten dengan data observasi sekaligus meminimalkan kesalahan prediksi. Ide dasar SVR adalah mencari *hyperplane* (bidang pemisah) yang mempunyai deviasi maksimum dari setiap titik data. Model mencari fungsi yang dapat memberikan prediksi terbaik, dengan mempertimbangkan batas toleransi atau margin kesalahan. SVR dapat digunakan untuk memecahkan masalah regresi pada kumpulan data yang sangat kompleks. Peneliti [21] mengatakan bahwa SVR menggunakan formula pada persamaan (6).

$$y = f(x) = \omega \cdot \varphi(x) + b \qquad (6)$$

Dimana $\varphi(x)$ adalah fungsi pemetaan nonlinier, ω adalah vektor bobot, "·" adalah perkalian titik dalam ruang fitur.

3). *Random Forest* (RF)

Algoritma RF menciptakan pohon keputusan berdasarkan subset acak dari data yang diberikan. Pohon keputusan terdiri dari *internal node*, *root node*, dan *leaf node* [22]. Estimasi setiap pohon kemudian digunakan untuk menentukan solusi yang paling menguntungkan melalui pemungutan suara. Proses ini memerlukan pembuatan pohon keputusan untuk setiap subset, memperoleh keluaran prediksi dari setiap pohon, dan kemudian melakukan pemungutan suara pada setiap hasil prediksi. Prediksi akhir akan ditentukan berdasarkan hasil perolehan suara terbanyak. Berikut formula *Random Forest* pada persamaan (7) [23].

$$\bar{r}_p(X, L_p) = \mathbb{E}_y\left[r_p(X, y, L_p)\right] \qquad (7)$$

Simbol y adalah pengecualian bersyarat yang terkait dengan parameter acak X dan kumpulan data L_p. Ketergantungan estimasi dalam sampel akan dilambangkan dengan $\bar{r}_p(X)$. Variabel acak digunakan untuk menentukan bagaimana pemotongan berturut-turut dilakukan ketika membangun pohon individu, seperti pemilihan koordinat pemisahan dan posisi pemisahan [23].

4). *eXtreme Gradient Boosting* (XGB)

XGB merupakan implementasi teknik peningkatan yang menggabungkan beberapa model prediktif yang relatif lemah menjadi model yang lebih kuat. XGB *Regression* digunakan untuk memprediksi nilai numerik atau kontinu berdasarkan fitur dalam suatu dataset. Algoritma ini memaksimalkan performa model dengan meminimalkan kesalahan prediksi dan dapat menangani data yang kompleks dan berjumlah besar. Model XGB terdiri dari beberapa pohon keputusan. Setiap pohon melihat sisa dari pohon sebelumnya dan menggunakan algoritma *gradient* untuk menemukan pohon keputusan baru yang akan dibangun. Keputusan akhir dibuat secara bersama-sama oleh beberapa pohon keputusan, dan hasil dari semua pohon dirangkum sebagai hasil prediksi akhir. Algoritma XGB didefinisikan pada persamaan (8) [24]:

$$\hat{y}_a^{(b)} = \sum_{d=1}^{D} f_d(X_a) = \hat{y}_a^{(b-1)} + f_c(X_b) \qquad (8)$$

Dimana $\hat{y}_a^{(b)}$ adalah hasil prediksi sampel a setelah iterasi b, D adalah himpunan semua pohon keputusan, f_d adalah hasil prediksi pohon ke-d, X_a adalah sampel input ke-a, $\hat{y}_a^{(b-1)}$ merupakan hasil prediksi pohon ke-$(b-1)$, dan $f_b(X_a)$ merupakan hasil prediksi pohon ke- b.

5). *Gradient Boosting Machine* (GBM)

GBM adalah algoritma *machine learning* yang termasuk dalam kategori *ansambel learning*. GBM memprediksi nilai target

berdasarkan kombinasi beberapa model pembelajaran yang lemah dalam bentuk pohon keputusan sederhana. Langkah algoritma GBM [25] dirangkum pada Gambar 6, dimana data masukan adalah $(x, y)_{i=1}^{N}$, jumlah iterasi mengacu pada M, fungsi kerugian sebagai $\Psi(y, f)$, $\rho(.)$ adalah fungsi kerugian, dan pilihan model pembelajar dasar sebagai $h(x, \theta)$.

Algorithm 1. Gradient Boost Algorithm Steps

Initialize $(\hat{f}_o)$ with a constant
for$(t = 1)$ to M **do**
  Compute the negative gradient $g_t(x)$
  fit a new base-learner function $h(x, \theta)$
  find the best gradient descent step-size (ρ_t):
  $(\rho_t) : \arg min_\rho \sum_{i=1}^{N} \Psi[y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i \theta_t)]$
  update the function estimate:
  $\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)$
end for

Gambar 6. Langkah algoritma GBM. Sumber: Malik et al [25]

6). *Light Gradient Boosting Machine* (LGBM)

LGBM *Regression* merupakan salah satu jenis model regresi yang menggunakan algoritma *boosted tree*. Model ini menggunakan pohon keputusan yang semakin ditingkatkan (pohon yang dikuatkan) untuk membentuk model yang kuat dan akurat dalam memprediksi angka berdasarkan fitur tertentu. Rumus matematika LGBM [26] disajikan pada persamaan (9) hingga (12).

$$F = (\sum i \in \mathrm{L}\lambda)^2 / \left((\sum i \in \mathrm{L}\beta) + K\right) \quad (9)$$

$$L = (\sum i \in \mathrm{R}\lambda)^2 / \left((\sum i \in \mathrm{R}\beta) + K\right) \quad (10)$$

$$W = (\sum i \in \mathrm{I}\lambda)^2 / \left((\sum i \in \mathrm{I}\beta) + K\right) \quad (11)$$

$$G = 0.5 \cdot (F + L + W) \quad (12)$$

G dapat dilihat sebagai fungsi skor yang menilai kemahiran struktural struktur pohon, Variabel λ dan β menunjukkan nilai statistik dari gradien orde pertama dan kedua dari fungsi kerugian, dan untuk struktur pohon tertentu, sampel himpunan cabang kiri dan kanan masing-masing dilambangkan dengan L dan R.

7). *CatBoost*

CatBoost Regression merupakan algoritma *machine learning* yang dirancang khusus untuk tugas klasifikasi dan regresi. *CatBoost* dapat memproses data kategorikal tanpa memerlukan proses *one-hot coding* serta dapat mengatasi *overfitting* dan performa tinggi. Langkah yang digunakan oleh *CatBoost* dapat diuraikan sebagai berikut [20]:

1. Latih model m_1 dengan data, hitung $error_1 = [Y_i - Y_{\_pred}] \cdot f_1(x_i)$.
2. Latih model m_2 dengan $[X_i, error_1]$, hitung $error_2 = [Y_i - Y_{\_pred}] \cdot f_2(x_i)$.
3. Pada akhir langkah kedua, hitung $F_1(x_i) = f_1(x_i) + f_2(x_i)$.
4. Lalu latih model $F_1[X_i, error_2]$, hitung $error_3 = [Y_i - F_1(x_i)]$.
5. Selanjutnya latih model m_3 dengan $[X_i, error_3]$, $f_3(x_i) = error$.

Setiap langkah akan diulang hingga *loss* berkurang. Lalu model akhir menjadi $F_m(x) = F_{m-1} + \sum_{m=1}^{n} YF_m$.

2.5. Pengujian dan Evaluasi

Pengujian model akan memperoleh nilai R^2 *Score*. Selanjutnya melakukan evaluasi model menggunakan metrik *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE) dan *Root Mean Squared Error* (RMSE). R^2 *Score* atau *R-Squared Score* adalah ukuran seberapa dekat letak titik data dari nilai data yang diprediksi [27]. Nilai R^2 *Score* merupakan persentase dari total varians yang dijelaskan oleh model. R^2 *Score* dirumuskan dalam persamaan (19) [28]. MAE adalah rata-rata dari semua nilai absolut kesalahan prediksi pada setiap kumpulan data validasi [27]. MAE dirumuskan dalam persamaan (20) [29]. MSE adalah rata-rata dari semua nilai kuadrat kesalahan prediksi dan biasanya digunakan sebagai *loss function* untuk membantu model menyesuaikan kembali bobotnya dan menyesuaikan garis yang memiliki kesalahan prediksi lebih rendah [27]. MSE dirumuskan dalam persamaan (21) [30]. RMSE hampir mirip dengan MSE, hanya melakukan akar kuadrat dari MSE. Pada RMSE, nilai kesalahan dikuadratkan terlebih dahulu sebelum nilai mean dihitung, sehingga RMSE berguna ketika nilai kesalahan yang besar tidak diinginkan [27]. RMSE dirumuskan dalam persamaan (22) [31].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (19)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (20)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (21)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (22)$$

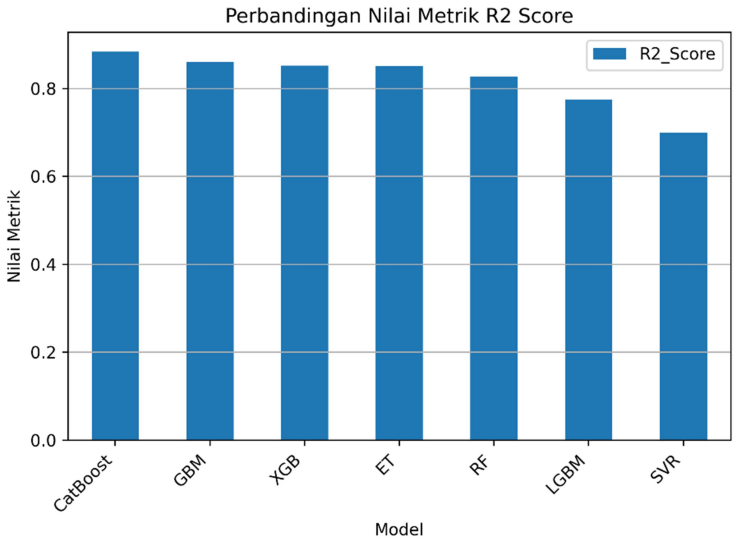
Keterangan:
y_i : nilai aktual
$\hat{y}_i$: nilai prediksi
n : jumlah sampel
$\bar{y}_i$: nilai rata-rata variabel target

3. HASIL DAN PEMBAHASAN

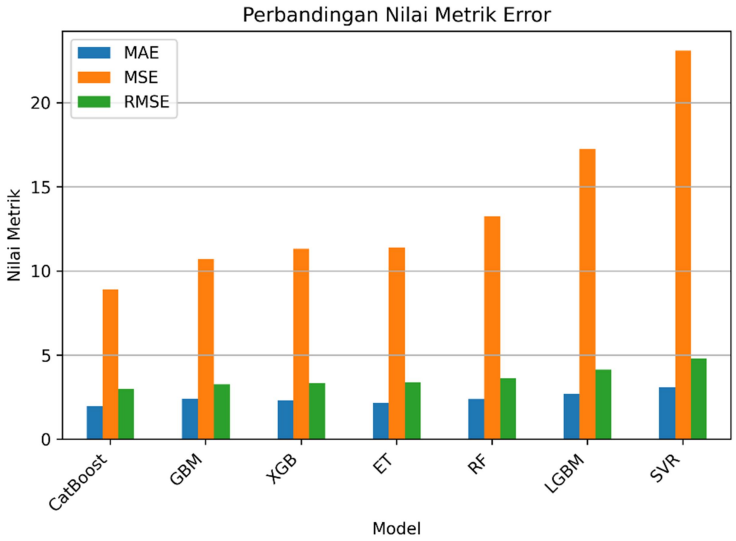
Pemrosesan awal kumpulan data melibatkan perbaikan data pada atribut RM, penskalaan dengan *Z-Score* dan PCA untuk reduksi dimensi data diterapkan pada penelitian ini. Tahap pengujian terbagi menjadi beberapa 4 bagian yaitu:

1. Pengujian tanpa PCA dan *GridSearchCV*.
2. Pengujian dengan PCA dan tanpa *GridSearchCV*.
3. Pengujian dengan GridSearchCV dan tanpa PCA.
4. Pengujian dengan PCA dan *GridSearchCV*.

Jumlah PC yang digunakan yaitu 6 PC dengan *cumulative explained variance* sebesar 0,9 dan 10 PC dengan *cumulative explained variance* sebesar 0,98. Pengujian dengan 10 PC untuk mengetahui apakah perbedaan 0,08 pada *cumulative explained variance* berpengaruh signifikan pada akurasi model. *GridSearchCV* (GSCV) adalah metode untuk mencari *hyperparameter* yang dapat meningkatkan suatu model dengan mencoba semua kombinasi hyperparameter secara menyeluruh sekaligus melakukan CV sebanyak 5 kali. *Cross Validation* (CV) adalah teknik yang digunakan untuk mengevaluasi kinerja model dengan cara membagi data menjadi beberapa subset yang saling bersinggungan. CV digunakan untuk membantu menghindari *overfitting* model pada data tertentu dan juga dapat membuat model menjadi lebih stabil ketika model diberikan data baru.



Gambar 7. Perbandingan nilai R^2 *Score* pada hasil pengujian tanpa PCA dan GridSearchCV.

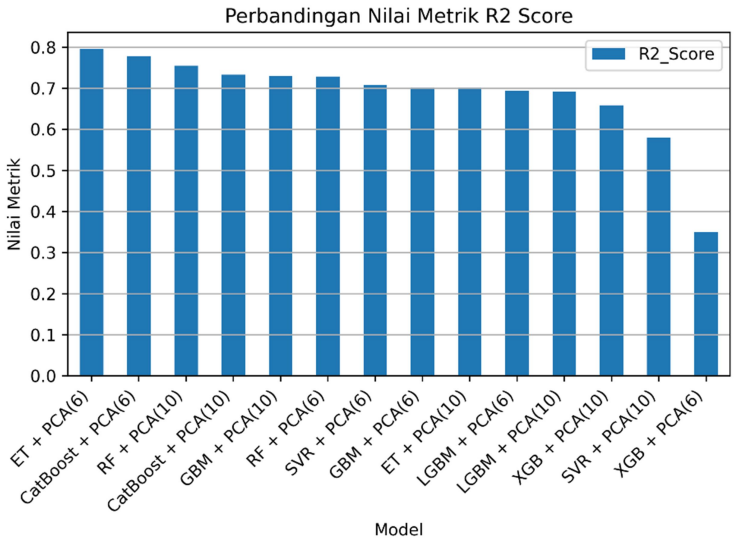


Gambar 8. Perbandingan nilai *Error* pada hasil pengujian tanpa PCA dan GridSearchCV.

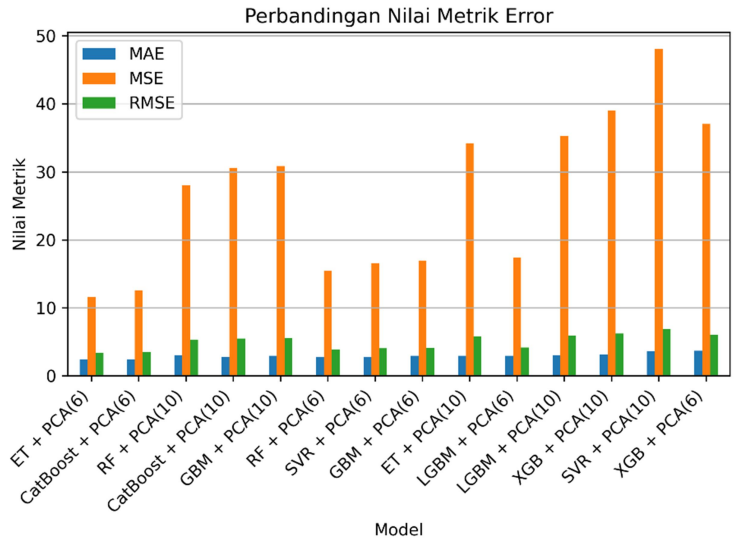
Tabel 2. Hasil pengujian tanpa PCA dan GridSearchCV

Metode	R^2	MAE	MSE	RMSE
CatBoost	0,883	1,995	8,908	2,984
GBM	0,860	2,405	10,719	3,274
XGB	0,852	2,334	11,324	3,365
ET	0,851	2,177	11,388	3,374
RF	0,827	2,394	13,253	3,640
LGBM	0,774	2,689	17,260	4,154
SVR	0,698	3,087	23,104	4,806

Pada Tabel 2 menunjukkan bahwa *CatBoost* memiliki akurasi yang lebih baik, sedangkan paling bawah ada SVR yang menggunakan kernel *linear* mendapatkan R^2 *Score* hanya sebesar 0,69. Pada pengujian bagian pertama ini hanya menggunakan *default parameter* dan tidak melakukan CV, berbeda dengan *GridSearchCV* yang mengoptimalkan *hyperparameter* sekaligus melakukan CV.



Gambar 9. Perbandingan nilai R^2 *Score* pada hasil pengujian dengan PCA dan tanpa GridSearchCV.

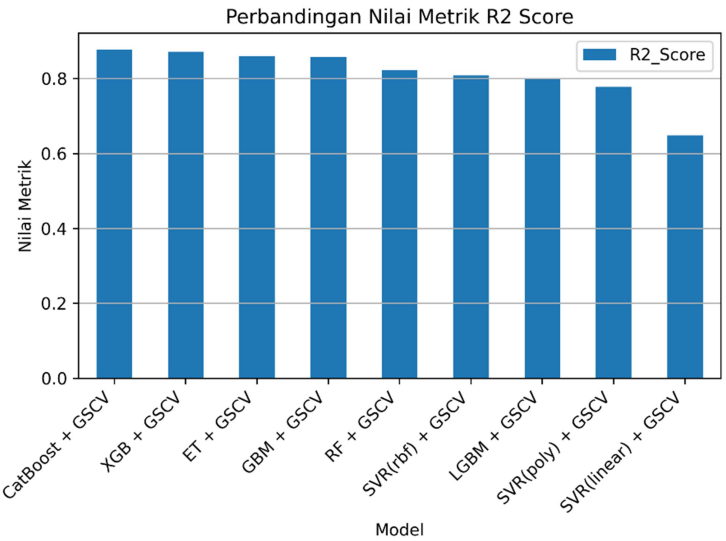


Gambar 10. Perbandingan nilai *Error* pada hasil pengujian dengan PCA dan tanpa GridSearchCV.

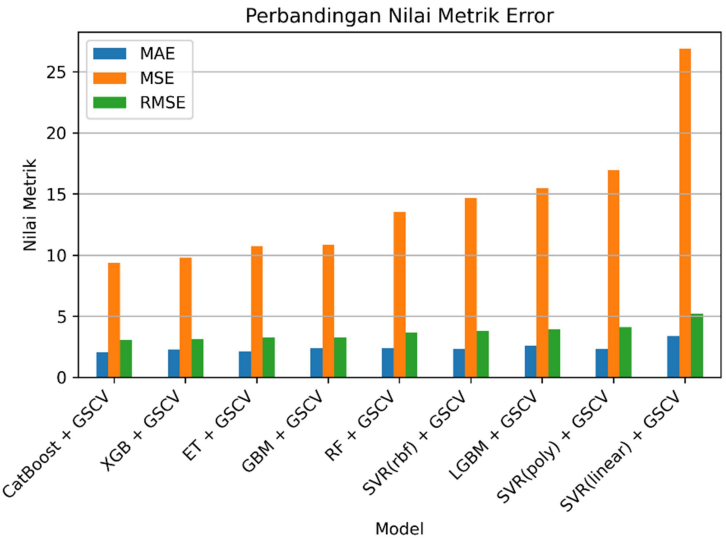
Tabel 3. Hasil pengujian dengan PCA dan tanpa GridSearchCV

Metode	R^2	MAE	MSE	RMSE
ET + PCA (6)	0,796	2,428	11,607	3,407
CatBoost + PCA (6)	0,778	2,449	12,615	3,551
RF + PCA (10)	0,755	3,081	28,060	5,297
CatBoost + PCA (10)	0,733	2,801	30,589	5,530
GBM + PCA (10)	0,730	3,005	30,870	5,556
RF + PCA (6)	0,729	2,764	15,442	3,929
SVR + PCA (6)	0,708	2,760	16,627	4,077
GBM + PCA (6)	0,702	2,980	16,962	4,118
ET + PCA (10)	0,701	2,956	34,213	5,849
LGBM + PCA (6)	0,694	2,987	17,405	4,172
LGBM + PCA (10)	0,692	3,090	35,276	5,939
XGB + PCA (10)	0,659	3,177	39,060	6,249
SVR + PCA (10)	0,580	3,667	48,110	6,936
XGB + PCA (6)	0,350	3,702	37,030	6,085

Pengujian bagian ke-2 (Tabel 3) tetap menunjukkan bahwa *CatBoost* menempati posisi pertama dengan R^2 *Score* 0,87, sedangkan SVR mengalami kenaikan dengan menggunakan kernel RBF, akan tetapi kernel *linear* tetap berada di posisi paling bawah. Hyperparameter yang digunakan oleh SVR kernel RBF yaitu C 90 dan *gamma auto*. Metode lainnya hanya mendapatkan sedikit peningkatan.



Gambar 11. Perbandingan nilai R^2 *Score* pada hasil pengujian dengan GridSearchCV dan tanpa PCA.

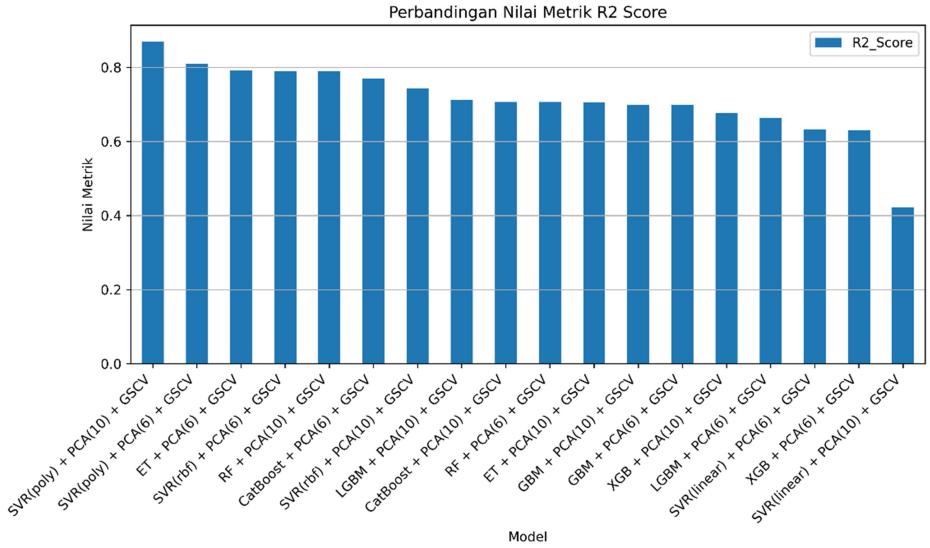


Gambar 12. Perbandingan nilai *Error* pada hasil pengujian dengan GridSearchCV dan tanpa PCA.

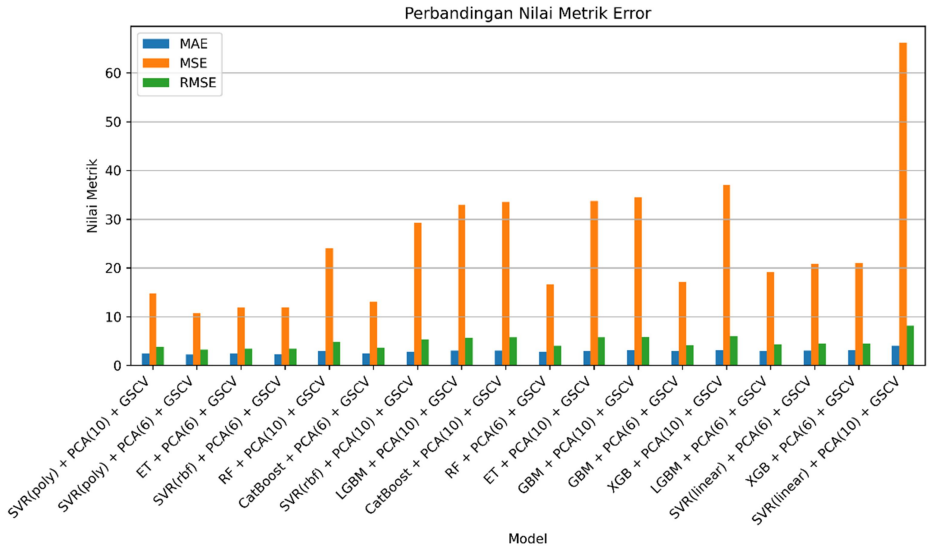
Tabel 4. Hasil pengujian dengan GridSearchCV dan tanpa PCA

Metode	R^2	MAE	MSE	RMSE
CatBoost + GSCV	0,877	2,045	9,362	3,059
XGB + GSCV	0,872	2,283	9,804	3,131
ET + GSCV	0,859	2,122	10,742	3,277
GBM + GSCV	0,858	2,382	10,835	3,291
RF + GSCV	0,823	2,381	13,543	3,680
SVR(RBF) + GSCV	0,808	2,339	14,675	3,830
LGBM + GSCV	0,798	2,602	15,495	3,936
SVR(*poly*) + GSCV	0,778	2,334	16,990	4,121
SVR(*linear*) + GSCV	0,649	3,369	26,910	5,187

Pengujian bagian ke-3 (Tabel 4) menggunakan PCA yang terbagi menjadi 2 yaitu 6 PC dan 10 PC. ET dengan 6 PC menempati posisi tertinggi, akan tetapi R^2 *Score* tidak lebih dari 0,80 yang dimana pengujian bagian ke-2 (lihat Tabel 3) terdapat 6 metode yang memiliki R^2 *Score* lebih dari 0,80. Enam PC yang hanya memiliki 0,90 *cumulative explained variance* berpengaruh lebih tinggi dibandingkan 10 PC dengan algoritma yang sama yaitu ET dan *CatBoost*, bahkan ET dengan 10 PC mengalami penurunan R^2 *Score* yang cukup tinggi dari 0,79 menjadi 0,70. Walaupun begitu, jumlah PC terlihat tidak terlalu berpengaruh pada CatBoost yang memiliki R^2 *Score* 0,77 menjadi 0,73, hanya berbeda 0,04.



Gambar 13. Perbandingan nilai R^2 *Score* pada hasil pengujian dengan PCA dan GridSearchCV.



Gambar 14. Perbandingan nilai *Error* pada hasil pengujian dengan PCA dan GridSearchCV.

Tabel 5. Hasil pengujian dengan PCA dan GridSearchCV

Metode	R^2	MAE	MSE	RMSE
SVR(*poly*) + PCA(10) + GSCV	0,870	2,430	14,837	3,851
SVR(*poly*) + PCA(6) + GSCV	0,809	2,239	10,849	3,293
ET + PCA(6) + GSCV	0,791	2,469	11,886	3,447
SVR(RBF) + PCA(6) + GSCV	0,790	2,368	11,938	3,455
RF + PCA(10) + GSCV	0,790	3,040	24,035	4,902
CatBoost + PCA(6) + GSCV	0,770	2,500	13,068	3,615
SVR(RBF) + PCA(10) + GSCV	0,744	2,776	29,264	5,409
LGBM + PCA (10) + GSCV	0,712	3,111	32,949	5,740
CatBoost + PCA(10) + GSCV	0,707	3,074	33,577	5,794
RF + PCA (6) + GSCV	0,706	2,856	16,706	4,087
ET + PCA (10) + GSCV	0,705	2,978	33,756	5,809
GBM + PCA (10) + GSCV	0,699	3,187	34,494	5,873

GBM + PCA (6) + GSCV	0,698	2,996	17,163	4,142
XGB + PCA (10) + GSCV	0,676	3,145	37,055	6,087
LGBM + PCA(6) + GSCV	0,664	2,957	19,137	4,374
SVR(*linear*) + PCA(6) + GSCV	0,633	3,118	20,884	4,569
XGB + PCA(6) + GSCV	0,631	3,163	21,030	4,585
SVR(*linear*) + PCA(10) + GSCV	0,421	4,097	66,262	8,140

Pengujian terakhir (Tabel 5) tetap menggunakan PCA yang terbagi menjadi 2 dan juga optimasi *hyperparameter* dengan *GridSearchCV*, Pada pengujian ini SVR dengan *kernel poly* dan 10 PC berada pada posisi tertinggi, hal ini sangat berbeda dibandingkan dengan pengujian bagian ke-3 dimana SVR 10 PC berada di posisi kedua dari bawah dengan *kernel default* yaitu *linear*. SVR *kernel poly* dengan 10 PC menggunakan *hyperparameter* C 0,5, *coef0* 5,0 dan *epsilon* 0,5 sedangkan SVR kernel poly dengan 6 PC yaitu C 0,5, coef0 1,0 dan epsilon 0,1. Hasil SVR *kernel poly* dengan metode PCA lebih memuaskan dibandingkan dengan tanpa PCA seperti pada Tabel 3 yang dimana SVR *kernel poly* hanya mendapatkan R^2 *Score* sebanyak 0,77, PCA dapat meningkatkan R^2 *Score* menjadi 0,87. Hasil SVR *kernel linear* selalu berada di bawah pada semua pengujian. Model RF beberapa kali mengalami peningkatan dan penurunan, walaupun begitu RF masih berada di 5 besar yang dapat mengindikasikan bahwa RF masih unggul dengan model lainnya. Hasil XGB menurun secara signifikan ketika memakai PCA, suatu hal yang berbeda ketika XGB menggunakan 13 atribut mendapatkan akurasi diatas 0,85. LGBM pada semua pengujian selalu mendapatkan R^2 *Score* yang tidak lebih dari 0,80. Hasil GBM tanpa PCA lebih baik jika dibandingkan dengan GBM 10 PC ataupun 6 PC, hal ini bisa disebabkan algoritma GBM membutuhkan kumpulan data yang utuh tanpa reduksi apapun, walaupun PCA adalah metode untuk mereduksi data dengan tujuan meningkatkan kinerja model, akan tetapi disisi lain PCA dapat menghilangkan beberapa informasi pada kumpulan data yang dapat menyebabkan algoritma GBM kekurangan informasi dan akhirnya menurun akurasinya. ET pada Tabel 5 dan Tabel 3 yang menggunakan 6 PC lebih unggul dibandingkan dengan 10 PC, hal ini bisa saja terjadi tergantung cara kerja algoritma, beberapa algoritma bisa mendapatkan hasil yang baik walaupun hanya beberapa PC atau atribut. Hal tersebut juga terjadi pada algoritma *CatBoost* seperti yang terlihat pada Tabel 5 dan Tabel 3 dimana hasil pengujian *CatBoost* dengan 6 PC lebih baik dari *CatBoost* 10 PC, walaupun hanya berbeda sedikit diantara keduanya.

Tabel 6. Perbandingan dengan penelitian sebelumnya

Author	Metode	R^2	MAE	MSE	RMSE
Chanasit et al. [32]	ANN 3 Layers + CV(5)	-	-	-	4,219
Chanasit et al. [32]	BIGA + CV(5)	-	-	-	3,673
Chanasit et al. [32]	RFE + CV(5)	-	-	-	3,737
Chanasit et al. [32]	RreliefF + CV(5)	-	-	-	3,899
Chanasit et al. [32]	MI + CV(5)	-	-	-	3,745
Penelitian ini	**CatBoost + GSCV**	**0,877**	**2,045**	**9,362**	**3,059**
Penelitian ini	**XGB + GSCV**	**0,872**	**2,283**	**9,804**	**3,131**
Penelitian ini	**SVR(*poly*) + PCA(10) + GSCV**	**0,870**	**2,430**	**14,837**	**3,851**
Penelitian ini	**ET + GSCV**	**0,859**	**2,122**	**10,742**	**3,277**
Penelitian ini	**GBM + GSCV**	**0,858**	**2,382**	**10,835**	**3,291**
Penelitian ini	**RF + GSCV**	**0,823**	**2,381**	**13,543**	**3,680**
Penelitian ini	**LGBM + GSCV**	**0,798**	**2,602**	**15,495**	**3.936**

Pada Tabel 6 merupakan perbandingan hasil pengujian yang dilakukan pada penelitian ini dengan penelitian sebelumnya, dataset yang digunakan sama dengan penelitian ini. Metode yang dibandingkan yaitu metode yang memiliki akurasi tinggi pada setiap algoritma, seperti metode RF yang memiliki akurasi lebih tinggi dibandingkan dengan metode RF+PCA(10)+GSCV, sehingga yang dimasukkan adalah metode RF. Penelitian [32] menggunakan metrik evaluasi yang berbeda dengan penelitian ini. Nilai RMSE yang lebih kecil menunjukkan bahwa metode tersebut memiliki kinerja yang lebih baik. Pada Tabel 6 terlihat bahwa *CatBoost* memiliki RMSE lebih kecil dibandingkan dengan metode lainnya. Keseluruhan perbandingan metode dengan penelitian sebelumnya dapat diketahui bahwa

CatBoost cocok untuk memprediksi harga *real estate*. Walaupun terdapat metode LGBM+GSCV yang memiliki RMSE sebesar 3,936, akan tetapi metode tersebut lebih baik dibandingkan dengan ANN 3 *Layers* dengan *Cross Validation* (CV) sebanyak 5 kali.

4. DISKUSI

Berdasarkan penelitian yang dilakukan menggunakan dataset yang diperoleh dari situs Kaggle dan melakukan berbagai tahap seperti pembersihan data, melakukan normalisasi dengan *Z-Score*, seleksi fitur dengan *Princical Component Analysis* (PCA) hingga penerapan *GridSearch Cross Validation* untuk memaksimakan kinerja algoritma *machine learning* ini. Hasil yang diperoleh menunjukkan bahwa metode *CatBoost* memiliki akurasi yang lebih tinggi dibandingkan metode lainnya bahkan mengalahkan metode yang digunakan oleh penelitian sebelumnya, sedangkan metode XGB dengan 6 PC berada di posisi paling bawah jika 4 bagian pengujian disatukan. Selain itu, CatBoost menjadi model yang stabil dalam 4 bagian pengujian tersebut. Walaupun begitu, hasil pengujian *CatBoost* yang R^2 *Score* mencapai 0,88 tidak dilakukan *cross validation*, sehingga rentan mengalami *overfitting* terhadap data baru. Pada Tabel 4 menunjukkan *CatBoost* yang sudah dilakukan CV sebanyak 5 kali dalam pencarian *hyperparameter* dengan GSCV tetap mendapat R^2 *Score* lebih tinggi dari model lainnya, sehingga akan mendapatkan hasil prediksi yang lebih stabil daripada sebelumnya.

Model SVR tidak stabil, bahkan ketika SVR *kernel Radial Basis Function* (RBF) mendapatkan R^2 *Score* sebesar 0,80 pada Tabel 4 menurun menjadi 0,74 pada pengujian selanjutnya (Tabel 5), tidak hanya itu *kernel* lainnya juga menunjukkan peningkatan ataupun penurunan secara signifikan.

Kinerja model tidak hanya ditentukan oleh penerapan metode PCA saja, akan tetapi bagaimana cara kerja algoritma tersebut, seperti XGB yang mengalami penurunan ketika menggunakan PCA. PCA dapat menyeleksi fitur yang penting tetapi terdapat beberapa fitur/informasi yang hilang, fitur/informasi yang hilang ini dapat memiliki pengaruh yang kuat terhadap beberapa cara kerja algoritma, sehingga metode PCA belum tentu cocok dengan semua algoritma. Metode GSCV sangat diperlukan dalam melakukan pengujian ini karena selain mencari *hyperparameter* yang dapat meningkatkan model, juga melakukan *Cross Validation* berguna untuk membantu model terhindar dari *overfitting* ketika diberikan data baru.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan model machine learning yang dapat memprediksi harga real-estate dengan akurasi tinggi berdasarkan beberapa faktor yang ada didalam dataset. Model ini menggunakan teknik PCA dan GridSearchCV pada algoritma SVR dengan kernel *poly* untuk mencapai R^2 *Score* yang tinggi sebesar 0,87. Namun pada pengujian GridSeacrhCV tanpa PCA, algoritma *CatBoost* memperoleh R^2 *Score* sebesar 0,883. Model ini diharapkan dapat memaksimalkan efisiensi dan kesederhanaan dalam memprediksi harga *real-estate* di masa depan.

DAFTAR PUSTAKA

[1] P.-F. Pai and W.-C. Wang, "Using Machine Learning Models and Actual Transaction Data for Predicting Real Estate Prices," *Applied Sciences*, vol. 10, no. 17, Sep. 2020, doi: 10.3390/app10175832.

[2] Q. Truong, M. Nguyen, H. Dang, and B. Mei, "Housing Price Prediction via Improved Machine Learning Techniques," *Procedia Comput Sci*, vol. 174, pp. 433–442, 2020, doi: 10.1016/j.procs.2020.06.111.

[3] R. Tanamal, N. Minoque, T. Wiradinata, Y. Soekamto, and T. Ratih, "House Price Prediction Model Using Random Forest in Surabaya City," *TEM Journal*, vol. 12, no. 1, pp. 126–132, Feb. 2023, doi: 10.18421/TEM121-17.

[4] O. Babb, "A Comparison of Machine Learning Approaches to Housing Value Estimation," *SIAM Undergrad Res Online*, vol. 12, Nov. 2019, doi: 10.1137/18S017296.

[5] J. Kang, H. J. Lee, S. H. Jeong, H. S. Lee, and K. J. Oh, "Developing a Forecasting Model for Real Estate Auction Prices Using Artificial Intelligence," *Sustainability*, vol. 12, no. 7, Apr. 2020, doi: 10.3390/su12072899.

[6] T. D. Phan, "Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia," *International Conference on Machine Learning and Data Engineering (iCMLDE)*, pp. 35–42, Jan. 2018, doi: 10.1109/iCMLDE.2018.00017.

[7] M. Tekin and I. U. Sari, "Real Estate Market Price Prediction Model of Istanbul," *Real Estate Management and Valuation*, vol. 30, no. 4, pp. 1–16, Dec. 2022, doi: 10.2478/remav-2022-0025.

[8] E. Tripathi, R. Shivaramakrishnan, D. Nanani, and A. Deshmukh, "Understanding Real Estate Price Prediction Using Machine Learning," *Int J Res Appl Sci Eng Technol*, vol. 9, no. 4, pp. 811–816, Apr. 2021, doi: 10.22214/ijraset.2021.33720.

[9] L. Rampini and F. R. Cecconi, "Artificial Intelligence Algorithms to Predict Italian Real Estate Market Prices," *Journal of Property Investment and Finance*, 2021, doi: 10.1108/JPIF-08-2021-0073.

[10] J. Kalliola, J. Kapočiūte-Dzikiene, and R. Damaševičius, "Neural Network Hyperparameter Optimization for Prediction of Real Estate Prices in Helsinki," *PeerJ Comput Sci*, vol. 7, pp. 1–25, Apr. 2021, doi: 10.7717/peerj-cs.444.

[11] M. Štubňová, M. Urbaníková, J. Hudáková, and V. Papcunová, "Estimation of Residential Property Market Price: Comparison of Artificial Neural Networks and Hedonic Pricing Model," *Emerging Science Journal*, vol. 4, no. 6, pp. 530–538, Dec. 2020, doi: 10.28991/esj-2020-01250.

[12] R.-T. Mora-Garcia, M.-F. Cespedes-Lopez, and V. R. Perez-Sanchez, "Housing Price Prediction Using Machine Learning Algorithms in COVID-19 Times," *Land (Basel)*, vol. 11, no. 11, Nov. 2022, doi: 10.3390/land11112100.

[13] M. Čeh, M. Kilibarda, A. Lisec, and B. Bajat, "Estimating the Performance of Random Forest Versus Multiple Regression for Predicting Prices of the Apartments," *ISPRS Int J Geoinf*, vol. 7, no. 5, May 2018, doi: 10.3390/ijgi7050168.

[14] F. Mostofi, V. Toğan, and H. B. Başağa, "Real-estate Price Prediction With Deep Neural Network and Principal Component Analysis," *Organization, Technology and Management in Construction*, vol. 14, no. 1, pp. 2741–2759, Jan. 2022, doi: 10.2478/otmcj-2022-0016.

[15] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data," *Information Fusion*, vol. 59, pp. 44–58, Jul. 2020, doi: 10.1016/j.inffus.2020.01.005.

[16] A. M. Siregar, J. H. Jaman, and A. Mufti, "Analisa Prediksi Kesehatan Masyarakat Indonesia Menggunakan Recurrent Neural Network," *INTERNAL (Information System Journal)*, vol. 4, no. 1, pp. 28–34, Jun. 2021, doi: 10.32627/internal.v4i1.285.

[17] E. V. P. Darshini, I. Vinuthna, G. B. S. Gayathri, G. Rani, and I. G. A. Roy, "Prediction of House Price Using Machine Learning Algorithms," *International Research Journal of Modernization in Engineering Technology and Science*, Mar. 2023, doi: 10.56726/irjmets34307.

[18] M. Heidari, S. Zad, and S. Rafatirad, "Ensemble of supervised and unsupervised learning models to predict a profitable business decision," in *2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021. doi: 10.1109/IEMTRONICS52119.2021.9422649.

[19] Koirunnisa, A. M. Siregar, and S. Faisal, "Optimized Machine Learning Performance with Feature Selection for Breast Cancer Disease Classification," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 4, pp. 1131–1143, 2023, doi: 10.26555/jiteki.v9i4.27527.

[20] K. S. R. Kundra, B. J. Lakshmi, I. V. S. Venugopal, and V. Guthula, "Flood Prediction using MLP, CatBoost and Extra-Tree Classifier," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 7 s, pp. 35–44, Jul. 2023, doi: 10.17762/ijritcc.v11i7s.6974.

[21] H. Han and W. Wang, "A Hybrid BPNN-GARF-SVR Prediction Model Based on EEMD for Ship Motion," *CMES - Computer Modeling in Engineering and Sciences*, vol. 134, no. 2, pp. 1353–1370, 2023, doi: 10.32604/cmes.2022.021494.

[22] A. Hidayanti, A. M. Siregar, S. A. P. Lestari, and Y. Cahyana, "Model Analisis Kasus Covid-19 Di Indonesia Menggunakan Algoritma Regresi Linier Dan Random Forest," *PETIR*, vol. 15, no. 1, pp. 91–101, Dec. 2021, doi: 10.33322/petir.v15i1.1487.

[23] T. L. Octaviani and Z. Rustam, "Random Forest for Breast Cancer Prediction," *AIP Conf Proc*, vol. 2168, Nov. 2019, doi: 10.1063/1.5132477.

[24] S. Guan, Y. Wang, L. Liu, J. Gao, Z. Xu, and S. Kan, "Ultra-short-term Wind Power Prediction Method Based on FTI-VACA-XGB Model," *Expert Syst Appl*, vol. 235, Jan. 2024, doi: 10.1016/j.eswa.2023.121185.

[25] A. Malik *et al.*, "Deep Learning Versus Gradient Boosting Machine for Pan Evaporation Prediction," *Engineering Applications of Computational Fluid Mechanics*, vol. 16, no. 1, pp. 570–587, 2022, doi: 10.1080/19942060.2022.2027273.

[26] S. Shi, "Comparison of Real Estate Price Prediction Based on LSTM and LGBM," *Highlights in Science, Engineering and Technology AMMSAC*, vol. 2023, 2023, doi: 10.54097/hset.v49i.8521.

[27] P. Kangane, A. Mallya, A. Gawane, V. Joshi, and S. Gulve, "Analysis of Different Regression Models for Real Estate Price Prediction," *International Journal of Engineering Applied Sciences and Technology*, vol. 5, no. 11, pp. 247–254, 2021, doi: 10.33564/IJEAST.2021.v05i11.041.

[28] M. Yazdani, "Machine Learning, Deep Learning, and Hedonic Methods for Real Estate Price Prediction," Oct. 2021, doi: 10.48550/arXiv.2110.07151.

[29] A. Georgiadis, "Real Estate Valuation Using Regression Models and Artificial Neural Networks: an Applied Study in Thessaloniki," *International Journal of Real Estate and Land Planning*, vol. 1, pp. 2623–4807, 2018, doi: 10.26262/reland.v1i0.6485.

[30] S. Khare, M. K. Gourisaria, H. GM, S. Joardar, and V. Singh, "Real Estate Cost Estimation Through Data Mining Techniques," *IOP Conf Ser Mater Sci Eng*, vol. 1099, no. 1, p. 012053, Mar. 2021, doi: 10.1088/1757-899x/1099/1/012053.

[31] C. Xue, Y. Ju, S. Li, and Q. Zhou, "Research on the Sustainable Development of Urban Housing Price Based on Transport Accessibility: a Case Study of Xi'an, China," *Sustainability (Switzerland)*, vol. 12, no. 4, Feb. 2020, doi: 10.3390/su12041497.

[32] K. Chanasit, E. Chuangsuwanich, A. Suchato, and P. Punyabukkana, "A Real Estate Valuation Model Using Boosted Feature Selection," *IEEE Access*, vol. 9, pp. 86938–86953, 2021, doi: 10.1109/ACCESS.2021.3089198.