

BAB II

LANDASAN TEORI

2.1 Objek Penelitian

2.1.1 Rumah

Rumah merupakan suatu bangunan dimana manusia tinggal dan melangsungkan kehidupannya. Rumah memiliki dua fungsi, pertama sebagai tempat tinggal dimana secara fisis orang bertempat tinggal, saat orang tersebut menemukan lingkungan alam yang cocok dan memiliki peralatan untuk bertempat tinggal. Kedua rumah sebagai tempat mediasi antara manusia dan dunia, mediasi memiliki arti adanya dialektik antara manusia dan dunia mereka [14].

Rumah adalah tempat berlindung utama untuk manusia dari iklim dan gangguan fisik lainnya. Rumah adalah suatu dasar dari pemeliharaan kemampuan produksi, tempat beristirahat, tempat yang digunakan untuk memelihara kesehatan, juga sebagai tempat belajar dan melakukan persiapan diri[15].

2.1.2 Lamudi

Lamudi merupakan perusahaan teknologi properti (PropTech) terbesar di Indonesia bersama dengan OLX Properti. Lamudi memiliki misi untuk membantu semua orang mendapatkan properti impian mereka dengan menghadirkan solusi end-to-end untuk perjalanan kepemilikan properti yang menghubungkan pengembang, agen, mitra bank, dan pencari properti dibawah satu platform yang mudah digunakan dan saling terpercaya.

Lamudi menjadi bagian dari grup teknologi properti yang besar, EMPG atau dengan kepanjangan *Emerging Markets Property Group* yang bertujuan untuk

1. **Business understanding**, Fase *business understanding* memfokuskan pada melakukan pemahaman pada tujuan dan kebutuhan dari sisi bisnis yang lalu akan melakukan ekstraksi informasi tersebut ke masalah dan terakhir melakukan perencanaan untuk mendapatkan tujuan tersebut.
2. **Data understanding**, Fase *data understanding* dimulai dengan melakukan pengambilan data dan proses melakukan pemahaman dan pendekatan terhadap data yang didapat. Fase ini juga dilakukan untuk melihat masalah-masalah yang ditemukan pada bagian kualitas data, mencari insight dari data ataupun mencari informasi tersembunyi dari data yang didapat.
3. **Data preparation**, Fase *data preparation* adalah fase dimana data mentah akan diproses dan dilakukan pembersihan sampai mendapatkan data final atau data yang akan dipakai nantinya.
4. **Modeling**, Fase *modeling* meliputi mengaplikasikan teknik *modeling* dan memilih mana yang terbaik dengan parameter terbaik untuk hasil performa yang optimal
5. **Evaluation**, Fase *evaluation* dilakukan dengan melakukan evaluasi lebih dalam pada model yang digunakan dan melakukan peninjauan lebih lanjut bahwa model dapat mencapai tujuan penelitian.
6. **Deployment**, Fase *deployment* adalah proses dimana model akan digunakan kepada *user* dengan membuat aplikasi ataupun *website* yang dapat digunakan *user*.

2.2.2 Random Forest

Random Forest adalah kombinasi dari *tree Predictor* yang pada setiap pohonnya memiliki ketergantungan pada nilai acak *vector* yang disampling secara independen dan dengan melakukan distribusi ke semua pohon yang ada pada *Forest* [18]. Breiman menemukan *random forest* yang berawal dari teknik

bagging dari penelitiannya sendiri dimana *random selection* dibuat untuk menumbuhkan setiap pohon pada model yang didapatkan dari data *training*.

Breiman juga memulai algoritma tersebut dari penelitian Dietterich pada tahun 1998 tentang *random split selection* yang pada setiap pembelahan *node* dipilih dengan acak pada *K best splits*. Breiman juga mendapatkan pada penelitiannya dengan membuat training set dengan mengacak *output* pada *training* set asli. Beberapa penelitian lainnya yang menginspirasi Breiman adalah penelitian Ho pada 1998 tentang *random* subspace dan juga penelitian Amit dan Geman pada tahun 1997 yang mendefinisikan banyak fitur geometric dan pencarian *random selection* dari *split* terbaik pada setiap *node*.

Random Forest menghasilkan ratusan atau bahkan ribuan *decision tree* yang akan bertindak sebagai fungsi regresi mereka sendiri dan pada *output* akhir dari *random Forest regression* adalah rata-rata dari *output* dari *decision tree* [19]. Setiap *decision tree* terdiri atas *decision nodes* dan juga *leaf nodes*. *Decision nodes* tersebut akan melakukan evaluasi pada setiap *sample* dengan fungsi tes dan melemparkan ke cabang lainnya berdasar dari fitur dari sampel tersebut. X direpresentasikan sebagai *input vector* yang berisikan m fitur dengan formula $X = \{x_1, x_2, \dots, x_m\}$, Y *output* scalar dan juga S_n pada data *training* yang berisikan n observation dengan formula $S_n = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}, X \in \mathbb{R}^m, Y \in \mathbb{R}$.

Model regresi *random Forest* merupakan hasil perluasan dari CART atau juga disebut *Classification and Regression Tree* dan dapat memberikan performa dari prediksi yang lebih baik. Menurut Antipov dan Elena, metode *random Forest* merupakan tipe metode dari *regression trees ensemble* yang memberi prediksi berdasar dari tiap pohon atau *tree* ataupun melakukan pemilihan dari vote terbanyak pada klasifikasi [20]. Semua *tree* pada *ensemble* dibuat secara independen dari algoritma dasarnya. Beberapa alasan *random Forest* menjadi teknik mass appraisal yang sangat sesuai yaitu:

1. Memiliki hasil yang baik dari beberapa jurnal-jurnal dan banyak penelitian menyebutkan bahwa *random Forest* sangat jarang memiliki masalah performa pada skema regresi
2. Dapat bekerja dengan kategorikal dari banyak tingkat. Misalnya, pada kasus *multiple regression* ataupun *neural network*, sejumlah besar *variable* menambahkan jumlah parameter yang biasanya membuat sebuah *overfitting*. Namun pada *random forest*, *variable* dengan *k categories* dicatat pada *k-1*, satu potongan dari pohon.
3. Dapat bekerja dengan *missing data*. Jika sejumlah data hilang, prediksi dilakukan dengan *tree* yang sudah dibuat dan tidak perlu mengecualikan observasi tersebut.
4. Dengan *bagging*, *random Forest* sangat tahan dari *outliers*
5. Pada satu pohon regresi, prediksi setiap objek adalah angka yang unik.
6. Dengan metode *regression trees* memperbolehkan *non-linear link* dan *variable* yang tidak stabil tetap mempengaruhi segmen lain
7. Metode *random Forest* tidak memerlukan spesifikasi model yang mendetail yang membuat pengiriman beberapa set berbeda dari determinan harga pada area yang berbeda
8. Prediksi untuk observasi baru pada jangkauan yang sama dianggap diproses seperti observasi yang sudah dilakukan yang menghindarkan dari *overestimation* ataupun *underestimation*
9. Pengukuran *factor importance* dengan mencari rata-rata pengurangan marginal pada residual jumlah *square* tiap variabelnya.

Random Forest berjalan dengan dua langkah yang pertama semua pohon akan menghasilkan *sequence instance* dan disampel secara acak dengan pengganti *training set* [10]. Setiap *sequence* dari *instance* akan berhubungan dengan *random vector* yang menggambarkan pohon-pohon tertentu. *Sequence*

akan sedikit lebih berbeda dari yang lainnya yang membentuk pohon-pohon tersebut. Prediksi dari pohon k dapat diformulakan seperti.

$$h_k(x) = h(x, \Theta) \forall k \in \{1, 2, \dots, K\} \quad (2.1)$$

dimana K merupakan jumlah *tree*. Setiap pohon akan memilih fitur secara acak untuk menghindari masalah korelasi. Ada banyak cara untuk membelah atau memecah *node* S menjadi dua subset, anggap threshold c dipilih sebagai fitur yang akan digunakan untuk membelah S menjadi S1, S2 yang didasari dari setiap nilai fitur v_i . Prediksi dari *subtree* didapat sebagai mean atau median *output* dari *instance* dengan decision rules yang sama. Prediksi akhir adalah dengan melakukan rata-rata pada tiap *output* pohon:

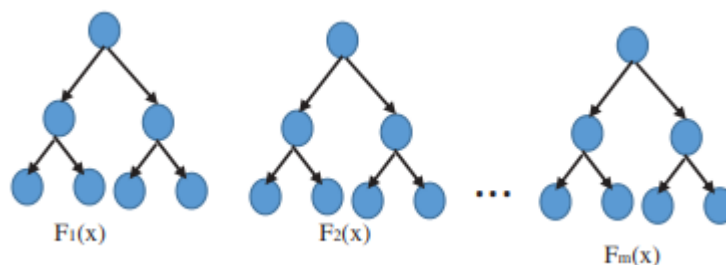
$$h(x) = \frac{1}{K} \sum_{i=1}^K h_k(x) \quad (2.2)$$

2.2.3 Gradient Boosting

Ide awal dari *Gradient Boosting* awalnya dikemukakan oleh Leo Breiman pada penelitiannya dimana mengatakan *boosting* memiliki *cost function* yang sesuai [21]. Selanjutnya, *Gradient Boosting* dikembangkan lebih lanjut oleh Jerome H. Friedman yang dimana ia menggunakan algoritma tersebut untuk prediksi dan klasifikasi [22]. Friedman juga menggambarkan *Gradient Boosting* yang mengonstruksikan model regresi dengan melakukan *fitting* secara sekuensial pada fungsi parameter sederhana ke “pseudo”-residual dengan *square* terkecil pada setiap iterasi.

Gradient Boosting merupakan *learning tools* yang sangat kuat, sebuah model *ensemble* dari *decision tree* yang diperuntukan untuk klasifikasi dan regresi [23]. Teknik *Gradient Boosting* dibuat dengan urutan pohon sederhana yang pada tiap pohon melakukan prediksi residual pada *tree* sebelumnya. *Gradient Boosting* merupakan prosedur *fitting* yang sangat bergantung pada

hyperparameter M [24]. Gambaran umum arsitektur dari *Gradient Boosting* dapat dilihat dari gambar dibawah



Gambar 2. 2 Gambar arsitektur *Gradient Boosting*

Gradient Boosting regression berbeda dari teknik *bagging* yang dimana melakukan metode secara sekuensial [25]. Akurasi dari prediksi ditingkatkan melalui pengembangan beberapa model pada sebuah urutan dengan memberi penekanan bahwa *training cases*nya yang sulit diestimasi. Perbedaan yang dapat terlihat dari *bagging* dan *boosting* adalah *boosting* memiliki strategi untuk melakukan *resample* pada data *training* untuk menyediakan informasi yang berguna pada tiap model. Distribusi dari tiap Langkah *training* berdasar dari *error* yang dibuat dari model sebelumnya. Tidak seperti *bagging* yang pada tiap *sample*-nya terbentuk untuk membuat *training dataset*, probabilitas dari pemilihan *sample* tidak sama pada *boosting*. Alur *Gradient Boosting* dimulai dengan [26]:

1. Melakukan inisialisasi prediksi dengan *decision tree* sederhana dengan formula

$$F_0(x) = \underset{y}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, y) \quad (2.3)$$

2. Melakukan kalkulasi residual yang adalah nilai dari prediksi dengan rumus

$$\tilde{y}_{im} = -\left[\frac{\partial \Psi(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad (2.4)$$

3. Membangun *decision tree* lain yang melakukan prediksi pada residual dari semua variable independen
4. Memperbarui prediksi dengan prediksi baru yang dikalkulasi dengan learning rate dengan formula seperti

$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} y_{jm} 1(x \in R_{J_m}) \quad (2.5)$$

5. Mengulang langkah 2 sampai ke 4 sesuai dengan jumlah iterasi atau jumlah *tree*

Gradient Boosting melakukan *fitting* pada model tambahan yang mengurangi fungsi rata-rata dari *training* data, seperti *square error* ataupun *absolute error*. *Loss function* tersebut mengukur banyaknya nilai yang diprediksi menyimpang dari nilai aslinya. Seperti pada penelitian Friedman, ia mengemukakan modifikasi *Gradient Boosting* dengan pohon regresi yang memiliki besar yang sudah ditetapkan seperti model dasarnya.

2.2.4 AdaBoost

AdaBoost yang juga dikenal sebagai *Adaptive Boosting* adalah metode *boosting* yang sangat dikenal karena seringnya meningkatkan akurasi *classifier* dengan melakukan rata-rata pada keputusan *ensemble* pada sebuah *classifier* [27]. *AdaBoost* membangun *classifier* dengan melakukan *training* secara sekuensial bersamaan dengan memasukkan penekanan yang memiliki pola tertentu.

AdaBoost memiliki teori properti yang sangat menarik dimana *AdaBoost* dapat menunjukkan *error* pada *classifier* yang komposit pada data *training* akan jatuh secara signifikan bersamaan dengan angka gabungan *classifier*

meningkat. Algoritma ini dinamakan *AdaBoost* karena tidak seperti algoritma-algoritma sebelumnya, *AdaBoost* melakukan penyesuaian terhadap *error* dari hipotesis lemahnya yang dikembalikan dengan *weak learn* [28]. *AdaBoost* dimulai dengan :

1. Membuat *weak learner* dengan beberapa variabel independen
2. Melakukan indentifikasi pada observasi yang salah diklasifikasi oleh *weak learner*.
3. Memperbarui beban observasi sesuai dengan klasifikasi yang salah pada *weak learner* sebelumnya akan diberi beban lebih dan klasifikasi yang benar pada *weak learner* sebelumnya diberi beban lebih sedikit
4. Penetapan beban pada tiap *weak learner* berdasar dari akurasi prediksi
5. Prediksi terakhir akan didapat dari beban rata-rata prediksi dari *weak learner*.

AdaBoost menyediakan analisis margin distribution yang dimana margin didefinisikan sebagai perbedaan antara skor *ensemble* pada *class* yang benar dan skor *ensemble* yang terkuat pada *class* yang salah. Dalam permasalahan ini terdapat dua kemungkinan label, $\{-1, +1\}$, yang merupakan $yf(x)$ dimana f adalah *output* dari *classifier* komposit dan y adalah label yang benar, klasifikasi dapat dikatakan benar jika margin adalah *positive*.

AdaBoost juga dapat meminimalkan hilangnya sebuah fungsi dan juga *squared error* yang terjadi. *AdaBoost* dapat dimengerti sebagai prosedur yang digunakan untuk meminimalkan sebuah *exponential loss* dimana $F(x)$ pada rumus dibawah. Dapat dilihat pada pilihan α_t dan h_t yang pada setiap ronde akan sama dengan yang dipilih seperti berkurangnya *loss* terbesar. Rumus dari *AdaBoost* dapat diformulakan seperti ini [29]:

$$F(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (2.6)$$

2.2.5 Flask

Flask merupakan *micro framework* pada *Python* yang menyediakan fungsi-fungsi dasar pada *framework* sebuah *web* dan memperbolehkan banyak *plugin* untuk dimasukkan ke fungsi dan fitur yang dapat diperluas ke tingkat selanjutnya[30]. *Flask* menggunakan *Jinja Template Engine* dan *Werkzeug WSG Toolkit*. *Flask* dikategorikan kepada dua bagian yaitu *Static files* dan *Template files*. Pada *template file* terdapat semua *jinja templates* seperti halaman *html*, dan pada *static file* berisikan kode statis yang diperlukan pada *website* seperti *css code*, *javascript* dan *file gambar*.

Flask disebut sebagai sebuah *micro framework* didasarkan dari fungsi utamanya yang sederhana tapi dapat diperluas pada sebuah pengembangan. *Flask* juga menjadi salah satu *micro framework* yang terpoles dan paling kaya dengan fitur yang ada sekarang [31]. *Flask* juga memiliki banyak manfaat seperti *fast templates*, fitur *WSGI* yang kuat, *testability* yang baik dalam *web* aplikasi dan tingkatan *library*, dan juga dokumentasi yang lengkap.

2.2.6 Cross validation

Cross validation merupakan metode *resampling data* yang sangat banyak digunakan untuk melakukan estimasi dari sebuah prediksi model dan melakukan peningkatan performa model dengan parameter [32]. *Cross validation* juga digunakan untuk melakukan generalisasi kemampuan prediksi dari model dan juga menghindari terjadinya *overfitting*. *Cross validation* awalnya digunakan untuk melakukan evaluasi validasi dari *linear regression* yang digunakan untuk melakukan ramalan dari *test baterai* [33].

2.2.7 R-squared

R-squared merupakan koefisien determinasi atau estimasi yang akan mengambil nilai dari sebuah *range* dari hubungan antar nilai asli dan juga hasil prediksi dari model [34]. *R-squared* memiliki *behavior independent* dari *linearity* dari model regresi dan dapat memiliki angka yang kecil pada *linear* model ataupun juga dapat menjadi besar pada model *non-linear*. *R-squared* memiliki formula sebagai berikut:

$$R^2 = 1 - \frac{MSE}{MST} \quad (2.7)$$

2.2.8 Root Mean Square Error (RMSE)

Root mean Square Error merupakan standar *metrics* pada pengukuran performa model yang digunakan untuk standarisasi dari pengukuran MSE. RMSE melakukan pengukuran dari keseluruhan akurasi pada model tapi akan meningkatkan *Prediction error rate* [35]. Formula pada RMSE adalah sebagai berikut:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_j - t_j)^2} \quad (2.8)$$

2.3 Tools dan Teknik

2.3.1 Python

Python merupakan *programming language* yang sangat mudah dipelajari dengan kapabilitas yang luas [36]. Menurut Paul F. Dubois dkk *Python* merupakan bahasa pemrograman yang digunakan sebagai *interpreter*. *Python* merupakan Bahasa *object-oriented scripting* yang sudah diinterpretasikan dan memiliki fitur yang sangat banyak dan berguna

seperti *function*, *garbage collection*, dan juga *exception handling*. *Python* sangat mudah diperluas dengan objek dan fungsi masing-masing dari penggunaannya.

Python merupakan bahasa pemrograman gratis tanpa harus memiliki lisensi untuk melakukan komersialisasi pada aplikasi penggunaannya. *Python* memiliki *library* yang sangat besar dengan modulnya. Modul ini dapat digunakan untuk banyak kebutuhan seperti audio dan image processing, pemrograman *web*, atau kebutuhan *user interface*. *Python* juga memiliki *array module* dengan tingkat performa yang tinggi seperti *Yorick*, *matlab*, *idl* ataupun *basis*.

2.3.2 Jupyter

Jupyter merupakan *software open source* yang dibuat untuk pembuatan *software* dan servis dengan standards pada interactive computing dari berbagai bahasa pemrograman [37]. Aplikasi utama dari *Jupyter* merupakan *Jupyter Notebook* merupakan aplikasi dengan format *open document* yang dapat digunakan *user* untuk membuat dan pembagian program mereka dan dapat dikombinasikan dengan narrative text, simbol ekuasi, visualisasi yang interaktif, gambar dan lain-lainnya.

Pada 2014, *Jupyter* muncul saat induk proyek mereka yang adalah *Ipython* yang digunakan untuk komputasi sains dan bahasa pemrograman *python* yang bergerak ke bidang data sains dan juga *machine learning*. Aplikasi lainnya dari *Jupyter* adalah *JupyterLab*, *nbconvert*, *Jupyter Widgets*, *Jupyter Hub*, *Binder*, *nbviewer* dan banyak lainnya.

2.3.3 Selenium

Selenium merupakan fitur atau *wrapper* pada *web* browser yang dapat membuat *web* browser seperti Google Chrome ataupun Internet Explorer yang dapat digunakan untuk melakukan otomasi dari proses browsing *website* dengan menggunakan bahasa pemrograman [38].

Selenium merupakan *software open source* yang dapat digunakan pada berbagai *OS* seperti *Windows*, *Linux*, ataupun *MacOS* dan dapat digunakan dengan komponen *WebDriver* mereka yang dapat melakukan automasi pada *web* browser [39]. Selenium juga dapat digunakan untuk melakukan *webscraping* dengan mengambil data-data ataupun *array* yang dapat ditemukan pada *web page*.

2.3.4 Google Data Studio

Google Data Studio merupakan sebuah program data visualisasi, didesain secara *user-friendly* yang digunakan untuk mempresentasikan data set kompleks secara menarik dan jelas [40]. *Google Data Studio* memiliki fungsi utama yang adalah penggambaran visual dengan gaya dashboard yang digunakan untuk penggambaran social media dan *web* analitik seperti *Google AdWord* dan *Youtube* tetapi *Google Data Studio* juga dapat mendukung penggunaan *tools* lain seperti *MySQL* dan *Google Sheets* yang membuat program *Google Data Studio* dapat digunakan peneliti untuk menggambarkan data mereka dengan visual yang menarik dan format yang *user-friendly*.

2.4 Penelitian Terdahulu

Tabel 2.1 Penelitian terdahulu

No	Nama Penelitian	Nama Jurnal	Nama Penulis	Hasil Penelitian
1	<i>A House Price Valuation Based On The Random Forest</i>	<i>International Journal of Strategic</i>	Jengei Hong, Choi Heeyoul	Hasil yang didapat pada penelitian tersebut adalah

	<i>Approach: The Mass Appraisal Of Residential Property In South Korea</i>	<i>Property Management Vol 24 Issue 3 (2020)</i>	, Kim Woo-sung	<i>random Forest</i> memiliki hasil yang lebih baik dibandingkan dengan prediksi OLS based dimana <i>random Forest</i> dapat mendapatkan hasil prediksi pada data yang kompleks dengan <i>r-squared</i> sebesar 0.97
2	<i>House Price Prediction Using Regression Techniques: A Comparative Study</i>	<i>IEEE 6th International Conference on smart structures and systems ICSSS (2019)</i>	C.H. Raga Madhuri, Anuradha G, dan M. Vani Pujitha	Penelitian tersebut menghasilkan bahwa <i>Gradient Boosting</i> memiliki akurasi yang paling besar yang sangat jauh jika dibandingkan dengan algoritma lainnya, <i>Gradient Boosting</i> mendapatkan hasil akurasi sebesar 0.917 dengan RMSE sebesar 10,971,390,390 .
3	<i>House Price Prediction using Random Forest Machine Learning Technique</i>	<i>Procedia Computer Science 199 (2022)</i>	Abigail Bola Adetunji dkk	Penelitian ini menggunakan data dari <i>dataset Boston housing</i> yang mengemukakan beberapa hasil prediksi harga dengan membandingkannya dengan harga yang sebenarnya dari data tersebut. Hasil dari penelitian ini adalah algoritma <i>random Forest</i> dapat melakukan prediksi harga rumah pada <i>dataset Boston housing</i> dimana pada

				perbandingan harga hanya memiliki nilai <i>difference</i> kurang lebih sebesar 5.
4	<i>Housing Prices Prediction with a Deep Learning and RandomForest Ensemble</i>	<i>Anais do XVI Encontro Nacional de Inteligência Artificial e Computaciona (2019)</i>	Bruno Klaus de Aquino Afonso dkk	Penelitian ini menghasilkan bahwa <i>random Forest</i> dengan teknik <i>Ensemble</i> dapat bekerja dengan baik dimana menghasilkan RMSLE sebesar 0.23847 dibandingkan dengan teknik lainnya.
5	<i>A new machine learning approach to house price estimation</i>	<i>New Trends in Mathematical Science Volume 4 Issue 6 (2018)</i>	Changchun Wang dan Hui Wu	Hasil yang didapat dari penelitian ini adalah algoritma <i>random Forest</i> jauh lebih efektif dan memiliki akurasi lebih besar jika dibandingkan dengan algoritma <i>linear regression</i> dimana pada <i>random Forest</i> memiliki akurasi sebesar 0.7 dan pada <i>linear regression</i> sekitar 0.4 pada model yang berisikan fitur <i>latitude-longitude</i> , tahun bangun dan juga luas tanah
6	PREDIKSI KELAYAKAN MASUK PENJURUSAN IPA SISWA SEKOLAH MENENGAH ATAS MENGGUNAKAN C4.5 (Studi Kasus:	Telematika Vol 10, No 2	Rasi Rahwali , Seng Hansun , Yustinus Widya Wiratama	Penggunaan cross-validation dengan implementasi data mining yang menghasilkan nilai tingkat error sebesar 16,65%

	SMA Tarakanita Gading Serpong)			
7	<i>Predicting the Case of COVID-19 in Indonesia using Neural Prophet Model</i>	IJNMT Vol 9, No 2	Efraim Yahya Wijaya, Alethea Suryadibrata	Prediksi kasus COVID-19 yang pada modelnya menghasilkan RMSE sebesar 5728.69 dan MAE sebesar 4007

Penelitian *A House Price Valuation Based On The Random Forest Approach: The Mass Appraisal Of Residential Property In South Korea* memiliki kontribusi terhadap referensi hasil random forest dan juga besar split data yang digunakan pada penelitian ini [7]. Penelitian *House Price Prediction Using Regression Techniques: A Comparative Study* memiliki peran dalam referensi hasil model Gradient Boosting dan AdaBoost yang dipakai dan juga menjadi acuan dalam nilai RMSE yang didapat [8]. *House Price Prediction using Random Forest Machine Learning Technique* memiliki peran penggunaan visualisasi data scatterplot hasil prediksi dengan nilai aktual [9]. *Housing Prices Prediction with a Deep Learning and RandomForest Ensemble* memiliki kontribusi dalam teori-teori metrics yang dipakai dalam penelitian ini hasil prediksi Random Forest yang didapat [10]. *A new machine learning approach to house price estimation* memiliki peran dalam inspirasi pemilihan Random Forest dan juga sebagai pertimbangan perbandingan linear regressionnya yang memiliki nilai yang kurang baik [11]. Penelitian PREDIKSI KELAYAKAN MASUK PENJURUSAN IPA SISWA SEKOLAH MENENGAH ATAS MENGGUNAKAN C4.5 (Studi Kasus: SMA Tarakanita Gading Serpong) memiliki peran dalam referensi dan inspirasi penggunaan cross-validation [12]. *Predicting the Case of COVID-19 in Indonesia using Neural Prophet Model* memiliki peran dalam referensi dan inspirasi hasil RMSE yang didapat dari penelitian tersebut untuk menjadi acuan dalam penelitian ini [13].

UNIVERSITAS
MULTIMEDIA
NUSANTARA