*TRANSMISI : JURNAL ILMIAH TEKNIK ELEKTRO, 24, (4), OKTOBER 2022*
*p-ISSN 1411-0814 e-ISSN 2407-6422*

# PERBANDINGAN KINERJA ALGORITME NAÏVE BAYES DAN K-NEAREST NEIGHBOR (KNN) UNTUK PREDIKSI HARGA RUMAH

Vania Ariyani Prilia Putri[*)], Agung Budi Prasetijo dan Dania Eridani

Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

[*)]*E-mail: vaniaariyani19@gmail.com*

## Abstrak

Rumah adalah bangunan yang berfungsi sebagai tempat tinggal/hunian dan sarana pembinaan keluarga, sehingga rumah merupakan salah satu kebutuhan dasar manusia. Seiring dengan berjalannya waktu, terjadi banyak perubahan yang berpengaruh terhadap kebutuhan akan rumah. Nilai–nilai dari setiap rumah pun beragam, seperti luas tanah, lokasi rumah, jumlah kamar, jumlah kamar mandi, luas ruang tamu, fasilitas yang ada di linkungan rumah, dan lain sebagainya. Dikarenakan adanya beragam nilai dari setiap rumah, hal tersebutlah yang membuat harga - harga rumah semakin bervariasi. Dengan *machine learning*, pembeli dapat memprediksi harga rumah dengan data rumah yang diberikan. Dalam pembuatan *machine learning* tersebut, dibutuhkan pembangunan model, dan selama proses pelatihan, diperlukan adanya suatu algoritma untuk membangun model yang disebut sebagai algoritma pelatihan (*learning algorithm*). Berdasarkan cara pelatihan, algoritma klasifikasi dibagi menjadi dua macam yaitu *eager learner* dan *lazy learner*. Namun, Dalam hal ini, peneliti termotivasi untuk melakukan analisis untuk membandingkan kinerja dari *eager learning* dan *lazy learning* dalam memprediksi harga rumah dikarenakan studi yang telah mengevaluasi dan membahas secara komprehensif kedua jenis pembelajaran tersebut masih sedikit.. Dalam penelitian ini, prediksi harga rumah dengan *machine learning* menggunakan algoritma Naïve Bayes sebagai perwakilan metode pembelajaran *eager learning* dan K-Nearest Neighbor mewakili metode pembelajaran *lazy learning*. Berdasarkan hasil penelitian yang dilakukan, model pembelajaran *lazy learning* memiliki kinerja yang lebih unggul dalam nilai *accuracy score* serta kecepatan waktu dalam proses *training data* dibandingkan model pembelajaran *eager learning*. Serta berdasarkan penelitian ini, kedua algoritma yang digunakan pada penelitian ini dapat dikatakan bahwa algoritma yang digunakan kurang bisa memprediksi harga rumah dengan baik, dikarenakan nilai *mean absolute error percentage* (MAPE) termasuk kategori "cukup", bukan "sangat baik".

*Kata kunci: Machine Learning, House Prices, Eager Learning, Lazy Learning, Naïve Bayes, K-Nearest Neighbor.*

## *Abstract*

*The house is a building that has functions as a place to live and a means of fostering a family, so that the house is one of the basic human needs. As time goes by, there have been many changes that affect the need for housing. The values of each house also vary, such as land area, location of the house, number of bedrooms, number of bathrooms, living room area, facilities in the home environment, and so on. Due to the various values of each house, this is what makes house prices more varied. With machine learning, buyers can predict house prices with the given home data. In making of machine learning, it is necessary to build a model, and during the training process, it is necessary to have an algorithm to build a model called a training algorithm (learning algorithm). Based on the training method, classification algorithms are divided into two types, namely eager learners and lazy learners. However, in this case, the researcher is motivated to conduct an analysis to compare the performance of eager learning and lazy learning in predicting house prices because there are still few studies that have comprehensively evaluated and discussed both types of learning. In this study, house price prediction using machine learning uses the Naïve Bayes algorithm as a representative of the eager learning learning method and K-Nearest Neighbor represents the lazy learning method. Based on the results of the research conducted, the lazy learning model has better performance in terms of accuracy score and speed of time in the data training process than the eager learning learning model. And based on this research, the two algorithms used in this study can be said that the algorithm used is less able to predict house prices well, because the mean absolute error percentage (MAPE) is in the "enough" category, not "very good".*

*Keywords: Machine Learning, House Prices, Eager Learning, Lazy Learning, Naïve Bayes, K-Nearest Neighbor.*

## 1. Pendahuluan

Dengan seiring berkembangnya zaman, kebutuhan manusia juga semakin meningkat. Namun, ada beberapa kebutuhan yang sangat diperlukan oleh manusia. Kebutuhan manusia yang wajib dipenuhi adalah kebutuhan primer, yang terdiri dari sandang (pakaian), pangan (makan), dan papan (tempat tinggal).

Tempat tinggal yang merupakan salah satu dari kebutuhan primer, berfungsi sebagai tempat untuk berteduh dan berlindung dari pengaruh eksternal manusia, seperti iklim, musuh, penyakit dan sebagainya. Contoh dari tempat tinggal yaitu rumah dan apartemen.

Tidak dapat dipungkiri, membeli rumah merupakan salah satu keputusan penting yang dapat diambil seseorang dalam hidupnya. Harga sebuah rumah mungkin tergantung pada berbagai macam faktor, mulai dari lokasi rumah, fitur, serta permintaan properti, dan permintaan pasar dan agen *real estate*[1].

Peningkatan teknik *machine learning* dan proliferasi data atau data besar yang tersedia, telah membuka jalan untuk studi *real estate* dalam beberapa tahun terakhir. *Machine Learning* yang merupakan salah satu cabang dari ilmu Kecerdasan Buatan, khususnya yang mempelajari tentang bagaimana komputer mampu belajar dari data untuk meningkatkan kecerdasannya[3]. *Machine learning* berarti menyediakan kumpulan data yang valid dan selanjutnya prediksi dilakukan berdasarkan data tersebut, *machine learning* juga mempelajari seberapa penting peristiwa tertentu yang mungkin dimiliki seluruh sistem berdasarkan data yang dimuat sebelumnya dan memprediksi hasilnya[4]. Pembangunan model selama proses pelatihan diperlukan adanya suatu algoritma untuk membangun model yang disebut sebagai algoritma pelatihan (*learning algorithm*). Berdasarkan cara pelatihan, algoritma klasifikasi dibagi menjadi dua macam yaitu *eager learner* dan *lazy learner*[5]. Sesuai dengan namanya, *Eager Learning* lebih banyak menginvestasikan waktu untuk fase *learning*, sedangkan *Lazy Learning* lebih banyak meluangkan waktu dalam fase klasifikasi[6].

Namun, studi yang telah mengevaluasi dan membahas secara komprehensif kedua jenis pembelajaran tersebut masih sedikit. Oleh karena itu, peneliti termotivasi untuk melakukan analisis untuk membandingkan kinerja dari *eager learning* dan *lazy learning* dalam memprediksi harga rumah. Dalam penelitian ini, akan dilakukan prediksi harga rumah dengan *machine learning* menggunakan algoritma Naïve Bayes dan K-Nearest Neighbor. Penulis memilih untuk menggunakan algoritma Naïve Bayes karena algoritma Naïve Bayes hanya membutuhkan sejumlah kecil data pelatihan untuk memperkirakan parameter yang diperlukan saat klasifikasi serta cepat dan efisien, sedangkan untuk algoritma K-Nearest Neighbor bisa digunakan untuk data kuantitatif maupun kualitatif, serta pelatihan yang sangat cepat, mudah untuk dipelajari, dan tahan terhadap data pelatihan yang memiliki *noise*

Dari penelitian ini, akan dianalisis hasil prediksi dan kinerja dari kedua algoritma tersebut. Dimana kedua algoritma tersebut mempunyai dua kategori yang berbeda, yaitu Naïve Bayes merupakan algoritma yang tergolong kedalam *Eager Learning*, dimana algoritma tersebut lebih banyak menginvestasikan waktu untuk fase *learning*. Sedangkan untuk K-Nearest Neighbor termasuk ke dalam kategori *Lazy Learning*, algoritma tersebut lebih banyak meluangkan waktu ke dalam fase klasifikasi[2].

Sebelum melakukan penelitian, penting untuk membuat acuan dari penelitian yang telah dilakukan sebelumnya. Hal tersebut digunakan untuk membandingkan dan menghubungkan hasil penelitian yang akan dilakukan dengan penelitian telah ada sebelumnya, hal tersebut dilakukan untuk menghindari duplikasi dari hal yang telah dilakukan, sehingga dapat diketahui kontribusi penelitian yang akan dilakukan ini dalam perkembangan penelitian.

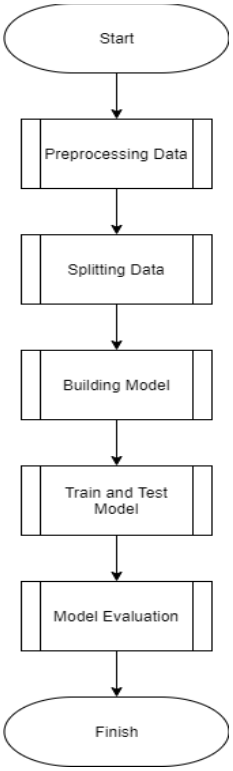
Pada penelitian berjudul "*Comparing lazy and eager learning models for water level forecasting in river-reservoir basins of inundation regions*"[14] oleh Chih-Chiang Wei, dilakukan penelitian dengan membandingkan model pembelajaran *lazy learning* dan *eager learning* untuk memprediksi ketinggian air dengan menggunakan algoritma *Locally Weighted Regression (LWR)* dan *K-Nearest Neighbor* (kNN) sebagai perwakilan *lazy learning*, lalu untuk *eager learning* digunakan algoritma *Artificial Neural Network* (ANN), *Support Vector Regression* (SVR), dan *Linear Regression* (REG). Pada penelitian didapatkan kesimpulan bahwa model pembelajaran *lazy learning* dan *eager learning* tidak menunjukkan keunggulan yang jelas antara satu sama lain, karena pendekatan model yang beragam, yang dapat dikategorikan sebagai model pembelajaran *lazy learning* atau *eager learning*, hal tersebut bergantung pada model itu sendiri[14].

Pada penelitian terdahulu mengenai *Machine Learning* yang berjudul "Sistem Prediksi Penyakit Kanker Serviks Menggunakan CART, *Naive Bayes*, dan K-NN"[2] membahas tentang klasifikasi penyakit kanker serviks menggunakan *Classification And Regression Trees* (CART), *Naive Bayes*, dan *k-Nearest Neighbor* (k-NN) menggunakan formula *Confusion Matrix*, dengan menggunakan teknik pemecahan *dataset Holdout*. Data yang digunakan pada penelitian tersebut didapatkan dari data sampel pasien yang pernah melakukan tes *Pap Smear* di RSUD Kediri dan YKI Kabupaten Kediri. Dari penelitian tersebut, didapatkan kesimpulan bahwa Algoritma *Naive Bayes* dengan teknik probabilistik mampu dengan baik melakukan klasifikasi terhadap kasus positif dan negatif kanker serviks, serta menghasilkan

tingkat akurasi yang tinggi, serta Algoritma k-NN dengan teknik klasifikasi sederhana pada kasus prediksi penyakit kanker serviks tidak dapat bekerja secara baik pada klasifikasi kasus positif. Dari penelitian tersebut, juga dapat diketahui bahwa algoritma yang termasuk dalam *eager learner* (CART *Decision Tree*, *Naive Bayes*) memiliki kinerja yang lebih baik dibandingkan *lazy learner* (k-NN)[2].

## 2. Metode Penelitian

Metode yang digunakan pada penelitian ini akan menjelaskan langkah-langkah penelitian pada Perbandingan Kinerja Algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) Untuk Prediksi Harga Rumah. Langkah atau tahapan yang dilakukan pada penelitian ini digambarkan melalui Gambar 1:




**Gambar 1. Tahapan Penelitian**

Data yang digunakan pada penelitian ini diambil dari dataset pada Kaggle dengan judul "*House Price Prediction*" yang dibuat oleh Shree, seorang data *scientist* dari Australia, dataset ini memiliki tag *Public* yang artinya dataset ini bebas digunakan sehingga akan terhindar dari *copyright* dari data yang digunakan. Dataset ini memiliki ekstensi csv yang berisi d*ate, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, sqft_above, sqft_basement, yr_built, yr_renovated, street, city, statezip, dan country.* Data *real estate* ini diambil dari data harga rumah yang ada di Amerika Serikat pada tahun 2014 dengan jumlah data sebanyak 4.601 data dari dataset pada Kaggle dengan judul "*House Price Prediction*". Parameter yang dipakai dalam memprediksi harga rumah yaitu *price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, sqft_above, city, statezip, day, month, basement, situation*, dan *renewal status.*

### 2.1. *Preprocessing Data*

Langkah pertama yang peneliti lakukan adalah *preprocessing data* atau biasa disebut dengan *preparation data. Preprocessing data* sendiri merupakan proses mempersiapkan data seperti membersihkan data dari *noise* maupun merubah format data [8]. Dalam proses ini, perlu membuat file berekstensi .ipynb pada *Google Colaboratory Playground.* Dengan membuat file .ipynb pada *Google Colaboratory* dapat memudahkan penelitian karena tidak diperlukan untuk meng*install Jupyter Notebook* atau *library – library* yang dibutuhkan. Selain itu untuk penyimpanan file–file seperti dataset penelitian akan tersimpan pada *Google Drive.*

Setelah selesai mempersiapkan *Google Colaboratory Playground*, proses *preprocessing data* dapat dilakukan. Data yang akan digunakan akan melalui beberapa tahap *preprocessing* seperti mengecek *missing data*, mengecek korelasi antar *feature*, menghapus *feature* yang mempunyai korelasi kecil dengan *feature price*, dan melakukan *feature scaling* dengan *standardization*[12].

### 2.2. *Splitting Data*

Langkah selanjutnya yaitu *splitting data*. *Splitting data* berfungsi untuk membagi *dataset* menjadi dua bagian, yaitu data *train* dan data *test* dengan proporsi tertentu. Nantinya data *train* akan digunakan sebagai data latih untuk membuat prediksi pada *machine learning* dan data *test* digunakan untuk mengevaluasi hasil fit model yang dibuat. Dengan melakukan *splitting data* akan memudahkan penelitian dalam melakukan prediksi, hal itu dikarenakan data yang digunakan sudah statis sehingga pada saat evaluasi hasil dengan data *test*, hasilnya tidak akan berubah – ubah.

### 2.3. *Building Model*

Setelah melakukan *splitting data*, langkah selanjutnya yaitu *building model* atau pembuatan model. Proses pembuatan model merupakan tahapan yang paling penting pada penelitian ini, karena proses prediksi memerlukan perancangan model untuk membaca dan mempelajari data. Pembuatan model akan menggunakan algoritma K- Nearest Neighbor dan Naïve Bayes. Pada pembuatan model juga akan dilakukan *fine tuning* untuk mengatur *active layer* pada masing – masing model. Untuk menghasilkan hasil penelitian yang optimal, dalam penelitian ini menggunakan *Grid Search* untuk menemukan *hyperparameter* terbaik, hal ini berdasarkan

penelitian yang dilakukan oleh Sateesh Ambesange dalam penelitian "*Liver Diseases Prediction using KNN with Hyper Parameter Tuning Techniques*" yang menyatakan *grid search* dapat menemukan kombinasi nilai parameter yang dapat memberikan hasil yang lebih baik dalam pengukuran akurasi performa[10].

### 2.4. Train and Test Model

Langkah selanjutnya yaitu *train* and *test* model. Pada proses ini model akan dilatih menggunakan data *train* yang sebelumnya sudah di *split* dari *dataset* yang digunakan. Setelah model selesai melakukan *training data*, kemudian model akan dievaluasi menggunakan data *test* untuk melihat hasil dari *training data* yang dilakukan sebelumnya.

### 2.5. Model Evaluation

Setelah *dataset* berhasil dilatih, langkah selanjutnya yaitu *model evaluation* atau evaluasi model. Pada tahap ini hasil *training* dari data *train* dan hasil *testing* dari data *test* akan dievaluasi untuk melihat kinerja dari masing-masing algoritma yang digunakan. Untuk mengevaluasi kinerja dari model bisa dilakukan dengan membandingkan nilai akurasi pada setiap model dan membandingkan waktu yang dibutuhkan dalam proses *training data* dengan pembagian komposisi jumlah data *train* dan data *test* yang berbeda.

Pada penelitian ini, variasi yang diberikan yaitu dengan memberikan sembilan variasi pada pembagian komposisi data *train* dan data *test*, sehingga dari variasi percobaan tersebut dapat dianalisis kinerja dari kedua algoritma yang digunakan pada penelitian. Komposisi data ini dibuat berdasarkan dari penelitian oleh Qisthina Syifa Setiawan pada "*Comparison of Naive Bayes and Decision Tree for Classifying Hepatocellular Carcinoma (HCC)*" yang membuktikan dari sembilan variasi komposisi data *train* dan data *test* didapatkan hasil nilai akurasi tertinggi sebesar 98,25% dengan pembagian data *train* dan data *test* sebesar 30:70[11], dan berdasarkan penelitian tersebut peneliti ingin membuktikan apakah dengan pembagian komposisi data yang sama akan menghasilkan tingkat akurasi yang sama dengan penelitian tersebut.

## 3. Hasil dan Analisis

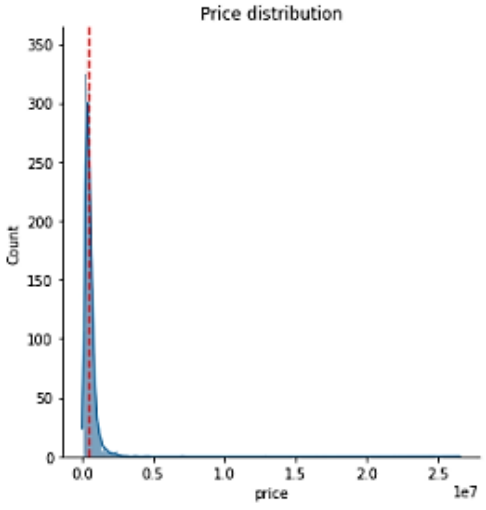
Pada bagian ini, akan membahas secara rinci hasil dari penelitian yang melalui beberapa tahapan seperti *preprocessing data, splitting data, building model, train and test model*, dan *model evaluation* dengan beberapa variasi jumlah data *train* dan data *test* yang berbeda. Pada akhir bab ini dilakukan perbandingan hasil prediksi terhadap kedua algoritma yang digunakan untuk melihat kinerja algoritma terbaik dari penelitian ini.

### 3.1. Pre-Processing Data

Sebelum melakukan prediksi, data yang akan digunakan perlu untuk melewati tahap pre-*processing data* terlebih dahulu, hal ini karena data yang dikumpulkan masih merupakan *raw data* yang perlu diubah menjadi informasi yang lebih bersih dan dapat digunakan pada tahap selanjutnya. *Dataset* yang digunakan pada penelitian berisi kolom *date*, *price*, *bedrooms*, *bathrooms*, *sqft_living*, *sqft_lot*, *floors*, *waterfront*, *view*, *condition*, *sqft_above*, *sqft_basement*, *yr_built*, *yr_renovated*, *street*, *city*, *statezip* dan *country*. Pertama, untuk mengetahui gambaran data yang akan digunakan, peneliti melihat informasi dari setiap kolom seperti tipe data dan jumlah total penggunaan memori yang digunakan, serta melihat apakah terdapat null data atau tidak.

Dari informasi data yang ditampilkan, diperlukan untuk mengubah tipe data dari beberapa kolom seperti kolom *date, price, bedrooms, bathrooms* dan *floors* karena data-data tersebut memiliki tipe data yang berbeda dengan data yang lainnya. Dikarenakan tahun pada setiap data adalah sama, sehingga tidak mempengaruhi prediksi, maka kolom *date* akan dibagi menjadi kolom *day*, *month*, dan *year*. Lalu, untuk kolom *date* dan *year* akan dihilangkan dari *dataset*.

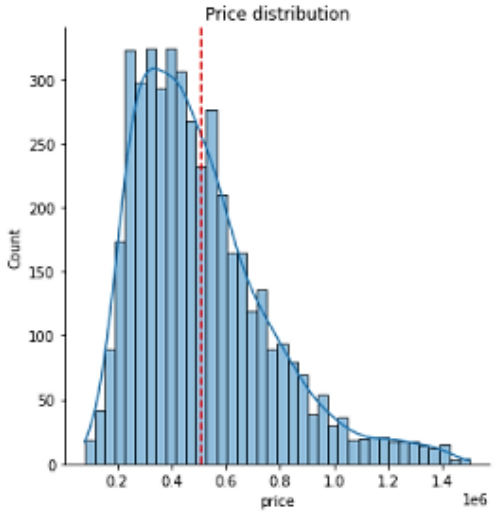
Setelah mengubah tipe data dari beberapa data dan menghilangkan informasi *year*, selanjutnya peneliti memvisualisasi pendistribusian harga rumah. Untuk grafik distribusi harga rumah bisa dilihat pada Gambar 2 Distribusi Harga Rumah.




**Gambar 2. Distribusi Harga Rumah**

Pada Gambar 2 Distribusi Harga Rumah dapat terlihat grafik distribusi harga rumah terlalu menumpuk ke kiri, hal ini disebabkan oleh data harga rumah yang harganya kurang dari 5.000.000 USD lebih banyak dibandingkan data yang memiliki harga rumah lebih dari 5.000.000

USD, sehingga menyebabkan persebaran data harga rumah tidak merata, dan hal ini dapat menghasilkan pengklasifikasian harga rumah yang kurang maksimal[13], sehingga peneliti memutuskan untuk meratakan distribusi harga rumah dengan hanya menggunakan data rumah yang mempunyai kisaran harga antara 20.000 USD – 1.500.000 USD agar saat diprediksi harga rumahnya menjadi lebih seimbang. Untuk grafik harga rumah dengan kisaran harga 20.000 USD – 1.500.000 USD dapat dilihat pada Gambar 3. Distribusi Harga Rumah dengan *range* Harga 20.000 USD - 1.500.000 USD.




**Gambar 3. Distribusi Harga Rumah dengan Range Harga 20.000 USD - 1.500.000 USD**

Pada data yang digunakan terdapat data *sqft_basement* yang menunjukkan luas dari *basement* yang dimiliki setiap rumah, *yr_built* yang berfungsi menunjukkan informasi tahun rumah tersebut dibangun, dan yr_*renovated* yang menunjukkan tahun rumah tersebut direnovasi. Dari beberapa data tersebut, peneliti memutuskan untuk menormalisasi data-data tersebut agar saat datanya diprediksi bisa menghasilkan hasil yang optimal. Normalisasi yang dilakukan peneliti adalah dengan memberikan *value* 1 untuk rumah yang memiliki *basement* dan 0 untuk rumah yang tidak memiliki *basement*. Selanjutnya, peneliti memberikan *value* 1 untuk rumah yang dibangun setelah tahun 1990 dan untuk rumah yang dibangun sebelum tahun 1990 diberi *value* 0. Setelah itu untuk rumah yang sudah pernah melakukan renovasi diberi *value* 1 dan untuk rumah yang belum pernah melakukan renovasi diberikan *value* 0. Nama kolom untuk sqft_basement, *yr_built*, dan *yr_renovated* diubah menjadi *basement, situation*, *renewal_status* untuk menggambarkan gambaran data yang telah dinormalisasi. Dikarenakan data dari kolom *sqft_basement*, *yr_built*, dan *yr_renovated* telah digantikan dengan data yang sudah dinormalisasi, maka data-data tersebut harus dihilangkan dari *dataset*. Setelah itu peneliti melakukan label *encoder* untuk mengubah data teks pada kolom *city* dan *statezip* menjadi numerik, dan peneliti melakukan *scaling data* terhadap *data price* yang menstandardisasi fitur dengan menghapus *mean* dan menskalakan ke varians unit[9].

### 3.2. *Splitting Data*

*Splitting data* adalah metode yang digunakan untuk membagi *dataset* menjadi data *train* dan data *test*. Fungsi dari data *train* yaitu data yang akan digunakan untuk dilatih pada saat pembuatan model, dan fungsi dari data *test* yaitu untuk data yang akan diuji dari hasil prediksi yang dibuat berdasarkan data *train*. Pada tahap *splitting data*, peneliti memutuskan untuk membuat sembilan variasi pembagian data *train* dan data *test*. Variasi yang digunakan pada penelitian ini yaitu 10:90, 20:80, 30:70, 40:60, 50:50, 60:40, 70:30, 80:20, 90:10 dengan berturut-turut data *train*: data *test*. Peneliti juga menyisakan data sebanyak 35 data yang disebut dengan *untouch data*, yang nantinya akan digunakan untuk melihat hasil prediksi harga rumah dari model yang telah dibuat.

### 3.3. Implementasi

Setelah *dataset* selesai di*split*, peneliti melanjutkan penelitian ke tahap selanjutnya, yaitu membangun model atau disebut juga *building model.* Pada tahap ini, peneliti menggunakan *grid search* untuk mencari *hyper parameter* yang akan digunakan pada saat menerapkan model ke *dataset* yang sudah dibagi. Untuk hasil pencarian *hyperparameter* menggunakan *grid search* akan ditunjukkan pada Table 1 Hasil Pencarian *Hyperparameter*.

**Table 1. Hasil Pencarian Hyper Parameter**

| Variasi Data *Train* dan Data *Test* | *Hyperparameter* |
|---|---|
| 10 : 90 | 39 |
| 20 : 80 | 175 |
| 30 : 70 | 171 |
| 40 : 60 | 194 |
| 50 : 50 | 56 |
| 60 : 40 | 119 |
| 70 : 30 | 197 |
| 80 : 20 | 171 |
| 90 : 10 | 94 |

*Hyperparameter* yang telah ditemukan dapat membantu menemukan kombinasi nilai parameter yang dapat memberikan hasil yang lebih baik dalam pengukuran akurasi performa[10]. Setelah menemukan *hyperparameter* pada setiap variasi data, nilai *hyperparameter* yang telah ditemukan diterapkan pada algoritma K-Nearest Neighbor. Dalam memprediksi harga rumah, algoritma K-Nearest Neighbor menggunakan *library* KNeighborsClassifier, sedangkan untuk Naïve Bayes menggunakan *library* GaussianNB. Pada penelitian ini, peneliti juga memberikan nilai *threshold* sebesar 5% untuk memberikan toleransi terhadap hasil dari harga rumah yang diprediksi.

### 3.4. Analisa Hasil Penelitian

Pada penelitian ini, peneliti meneliti dua hal, yaitu *accuracy score* dari masing – masing algoritma dan waktu yang dibutuhkan masing – masing algoritma dalam proses *training data*. Sehingga, pada penelitian ini didapatkan sembilan data *accuracy score* dan sembilan data waktu yang dibutuhkan dalam proses *training* data untuk setiap algoritma yang digunakan pada penelitian.

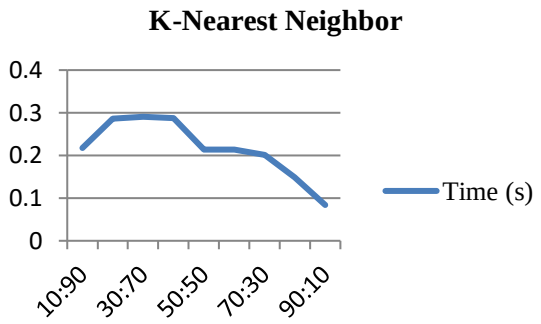
#### 3.4.1. Algoritma K-Nearest Neighbor

Pada penelitian algoritma K-Nearest Neighbor, peneliti menganalisis waktu yang dibutuhkan dalam *training* data pada kedua algoritma. Untuk besar waktu yang dibutuhkan algoritma K-Nearest Neighbor dalam *training data*, dapat dilihat pada Table 2 Waktu *Training Data* dengan Algoritma K-Nearest Neighbor. Dapat terlihat pada Table tersebut, variasi data *train* yang menghabiskan waktu paling sedikit dalam proses *training* data yaitu variasi data 90:10 dengan menghabiskan waktu sebanyak 0,0839 detik. Untuk variasi data yang menghabiskan waktu paling banyak dalam proses training data yaitu variasi data 30:70 dan menghabiskan waktu sebesar 0,2906 detik.

**Table 2. Waktu *Training Data* dengan Algoritma K-Nearest Neighbor**

| Variasi Data Train : Data Test | Time (s) |
|---|---|
| 10 : 90 | 0,2175 |
| 20 : 80 | 0,2858 |
| 30 : 70 | 0,2906 |
| 40 : 60 | 0,2872 |
| 50 : 50 | 0,2136 |
| 60 : 40 | 0,2139 |
| 70 : 30 | 0,2013 |
| 80 : 20 | 0,1483 |
| 90 : 10 | 0,0839 |

Bila melihat Gambar 4 Grafik Waktu yang dibutuhkan Algoritma K-Nearest Neighbor dalam *Training Data*, grafik tersebut menunjukkan kecenderungan yang menurun, dimana artinya semakin besar variasi data *train* yang diberikan maka semakin sedikit pula waktu yang dihabiskan dalam proses *training data*. Hal ini menunjukkan bahwa pernyataan metode pembelajaran *lazy learning* yang menghabiskan waktu lebih sedikit adalah benar, seperti pada jurnal penelitian Tampilan Klasifikasi Status Gunung Berapi dengan Metode *Learning Vector Quantization* (LVQ) oleh Virkhansa[7]. Hal ini dapat terjadi dikarenakan algoritma yang berbasis *lazy learning* menunggu data pengujian muncul, setelah data pengujian muncul baru dilakukan penyimpanan data yang nantinya akan digunakan untuk proses klasifikasi.



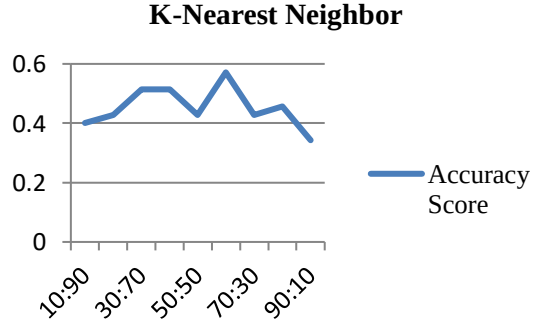

**Gambar 4. Grafik Waktu yang dibutuhkan Algoritma K-Nearest Neighbor dalam *Training Data***

Berikutnya, pada penelitian dengan algoritma K-Nearest Neighbor, didapatkan sembilan hasil *accuracy score* yang didapat dari memprediksi data rumah yang tidak disertakan pada proses *training* dan *testing*. Untuk hasil dari *accuracy score* algoritma K-Nearest Neighbor dapat terlihat pada Table 3. *Accuracy Score* K-Nearest Neighbor.

**Table 3. *Accuracy Score* K-Nearest Neighbor**

| Variasi Data Train : Data Test | Accuracy Score |
|---|---|
| 10 : 90 | 0,4 |
| 20 : 80 | 0,4285 |
| 30 : 70 | 0,5142 |
| 40 : 60 | 0,5142 |
| 50 : 50 | 0,4285 |
| 60 : 40 | 0,5714 |
| 70 : 30 | 0,4285 |
| 80 : 20 | 0,4571 |
| 90 : 10 | 0,3428 |

Seperti yang terlihat pada Table 3. *Accuracy Score* K-Nearest Neighbor yang mempunyai *accuracy* tertinggi adalah variasi data *train* dan data *test* sebesar 60:40, dimana 60% adalah data *train* dan 40% adalah data *test*, dan besar *accuracy score*nya yaitu 0,5714. Untuk variasi data *train* dan data *test* yang mempunyai *accuracy score* terendah yaitu variasi data 90:10, dimana 90% adalah data *train* dan 10% adalah data *test* yang mempunyai *accuracy score* 0,3428.




**Gambar 5. Grafik Accuracy Score dengan Algoritma K-Nearest Neighbor**

Bila melihat pada Gambar 5 Grafik *Accuracy Score* dengan Algoritma K-Nearest Neighbor, besarnya *accuracy score* tidak berpengaruh pada pertambahan besar variasi untuk data *train*, pasalnya pada variasi data *train* 10% hasil *accuracy score* yang didapatkan sebesar 0,4 dan pada variasi data *train* sebesar 20% hasil *accuracy score*nya meningkat 0,4285, serta pada vaiasi 30:70 dan variasi 40:60 menghasilkan *accuracy score* sebesar 0,5142, namun pada variasi data *train* sebesar 50% hasil *accuracy score*nya menurun ke angka 0,4285 yang berarti terjadi penurunan sebesar 0,0857. Lalu hasil *accuracy score* nya meningkat kembali pada variasi data 60:40 dengan hasil 0,5714.

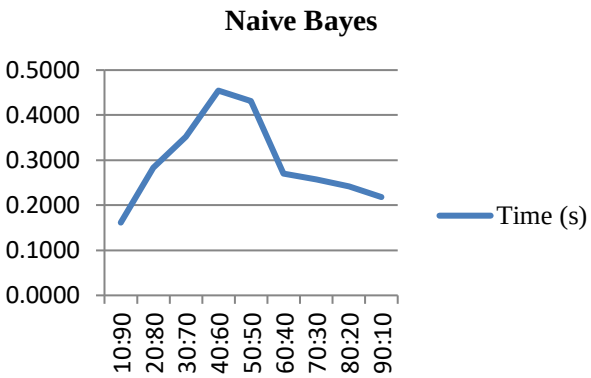
### 3.4.2. Algoritma Naïve Bayes

Pada algoritma Naïve Bayes, peneliti juga menganalisis waktu yang dibutuhkan dalam *training data* pada algoritma Naïve Bayes seperti pada penelitian dengan algoritma K-Nearest Neighbor. Untuk rincian besar waktu yang dibutuhkan algoritma Naïve Bayes dalam *training data*, dapat dilihat pada Table 4 Waktu *Training Data* dengan Algoritma Naïve Bayes.

**Table 4. Waktu *Training Data* dengan Algoritma Naive Bayes**

| Variasi Data Train : Data Test | Time (s) |
|---|---|
| 10 : 90 | 0,1615 |
| 20 : 80 | 0,2833 |
| 30 : 70 | 0,3516 |
| 40 : 60 | 0,4545 |
| 50 : 50 | 0,4318 |
| 60 : 40 | 0,2702 |
| 70 : 30 | 0,2575 |
| 80 : 20 | 0,2418 |
| 90 : 10 | 0,2184 |

Pada algoritma Naïve Bayes, waktu yang dibutuhkan dalam proses *training data* paling sedikit adalah proses *training* yang dilakukan pada variasi data 10:90 dan untuk variasi data yang membutuhkan waktu paling banyak pada proses *training data* yaitu variasi data 50:50. Jika melihat dari Gambar 6. Grafik Waktu yang Dibutuhkan Untuk *Training* dengan Algoritma Naive Bayes, waktu yang dibutuhkan dalam proses *training data* cenderung meningkat dari variasi data 10:90 hingga puncaknya di variasi data 50:50 dan ini sesuai dengan pernyataan bahwa algoritma dengan metode pembelajaran *eager learning* lebih banyak menginvestasikan waktu untuk fase *training* yang berkebalikan dengan *lazy learning* adalah benar, meskipun setelah variasi data 50:50 waktu yang dibutuhkan dalam proses *training data* cenderung menurun.

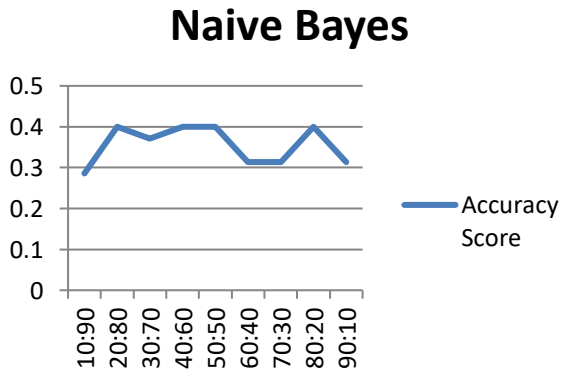



**Gambar 6. Grafik Waktu yang Dibutuhkan Untuk Training dengan Algoritma Naive Bayes**

Pada penelitian yang menggunakan algoritma Naïve Bayes, didapatkan sembilan hasil *accuracy score* yang didapat dari memprediksi data yang tidak disertakan pada proses *training* dan *testing* untuk setiap variasi data *train* dan data *test* yang berbeda seperti pada Table 5 *Accuracy Score* Naïve Bayes. Pada Table tersebut, dapat terlihat variasi data *train* dan data *test* yang menghasilkan *accuracy score* paling tinggi dengan nilai 0.4 adalah variasi data 20:80, 40:60, 50:50, dan variasi data 80:20.

**Table 5 Accuracy Score Naive Bayes**

| Variasi Data Train : Data Test | Accuracy Score |
|---|---|
| 10 : 90 | 0.2857 |
| 20 : 80 | 0.4 |
| 30 : 70 | 0.3714 |
| 40 : 60 | 0.4 |
| 50 : 50 | 0.4 |
| 60 : 40 | 0.3142 |
| 70 : 30 | 0.3142 |
| 80 : 20 | 0.4 |
| 90 : 10 | 0.3142 |




**Gambar 7 Grafik *Accuracy Score* Naive Bayes**

Jika diurutkan, maka hasil *accuracy score* dari yang tertinggi hinggga terendah adalah variasi data 20:80, 40:60, 50:50, 80:20, 30:70, 60:40, 70:30, 90:10, dan variasi data yang mempunyai *accuracy score* terendah adalah variasi data 10:90. Jika melihat Gambar 7 Grafik *Accuracy Score* Naïve Bayes, untuk hasil *accuracy score* dengan algoritma Naïve Bayes cenderung meningkat dengan semakin besarnya variasi data *train* yang diberikan, meskipun terdapat penurunan *accuracy score* pada variasi data 50:50 dan variasi data 80:20.

### 3.4.3. Rangkuman Analisa

Dari hasil penelitian yang didapatkan, peneliti mendapatkan hasil waktu yang digunakan dalam proses *training* dan nilai *accuracy score* dari masing–masing algoritma untuk setiap variasi data yang digunakan. Tahap ini merupakan rangkuman dari penelitian yang bertujuan untuk mempermudah melihat perbandingan dari kedua algoritma.

**Table 6. Perbandingan Kinerja Algoritma K-Nearest Neighbor dan Naïve Bayes**

| Variasi Data | Time(s) | | Accuracy Score | |
|---|---|---|---|---|
| | Naïve Bayes | KNN | Naïve Bayes | KNN |
| 10 : 90 | 0,1615 | 0.2175 | 0.2857 | 0.4 |
| 20 : 80 | 0.2833 | 0.2858 | 0.4 | 0.4285 |
| 30 : 70 | 0.3516 | 0.2906 | 0.3714 | 0.5142 |
| 40 : 60 | 0.4545 | 0.2872 | 0.4 | 0.5142 |
| 50 : 50 | 0.4318 | 0.2136 | 0.4 | 0.4285 |
| 60 : 40 | 0.2702 | 0.2139 | 0.3142 | 0.5714 |
| 70 : 30 | 0.2575 | 0.2013 | 0.3142 | 0.4285 |
| 80 : 20 | 0.2418 | 0.1483 | 0.4 | 0.4571 |
| 90 : 10 | 0.2184 | 0.0839 | 0.3142 | 0.3428 |

Dapat terlihat pada Table 6 Perbandingan Kinerja Algoritma K-Nearest Neighbor dan Naïve Bayes, pada penelitian ini algoritma K-Nearest Neighbor mempunyai kinerja yang lebih baik dibandingkan dengan algoritma Naïve Bayes dari segi waktu yang dibutuhkan dalam proses *training* maupun dari hasil *accuracy score*. Pada algoritma K-Nearest Neighbor didapatkan hasil *accuracy score* tertinggi dengan nilai 0,5714 pada variasi data 60:40, sedangkan untuk *accuracy score* tertinggi pada algoritma Naïve Bayes didapatkan oleh variasi data 20:80, 40:60, 50:50, dan 80:20 dengan nilai 0,4.

**Table 7. Kriteria Nilai MAPE**

| Nilai MAPE | Kriteria |
|---|---|
| <10 | Sangat Baik |
| 10-20 | Baik |
| 20-50 | Cukup |
| >50 | Buruk |

Pada penelitian ini, variasi data yang menghasilkan *accuracy score* tertinggi adalah variasi data *train* dan data *test* 60:40 dengan algoritma K-Nearest Neighbor. Berdasarkan prediksi dari variasi data *train* dan data *test* tersebut, peneliti mengukur nilai *error rate absolute* dengan rumus MAPE untuk mengukur seberapa tepat atau akurat suatu prediksi yang dihasilkan[8] dan ditunjukkan pada Table 8 Hasil Perhitungan MAPE untuk Variasi Data 60:40 dengan Algoritma K-Nearest Neighbor. Pada perhitungan tersebut, didapatkan nilai MAPEnya sebesar 43,52, yang mana nilai tersebut masuk ke kategori "cukup" dalam memprediksi harga rumah pada penelitian ini.

**Table 8. Hasil Perhitungan MAPE untuk Variasi Data 60:40 dengan Algoritma K-Nearest Neighbor**

| *Actual* | *Prediction* | $\lvert(\widehat{y}_i - y_i)/y_i\rvert$ |
|---|---|---|
| 464,600 | 285,000 | 0.39 |
| 132,250 | 395,000 | 1.99 |
| 433,111 | 345,000 | 0.20 |
| 542,500 | 325,000 | 0.40 |
| 368,112 | 330,000 | 0.10 |
| 673,476 | 210,000 | 0.69 |
| 558,653 | 300,000 | 0.46 |
| 168,333 | 210,000 | 0.25 |
| 268,971 | 330,000 | 0.23 |
| 318,000 | 375,000 | 0.18 |
| 550,607 | 550,000 | 0.00 |
| 584,000 | 345,000 | 0.41 |
| 245,000 | 475,000 | 0.94 |
| 287,919 | 210,000 | 0.27 |
| 672,500 | 330,000 | 0.51 |
| 454,790 | 300,000 | 0.34 |
| 282,508 | 400,000 | 0.42 |
| 473,200 | 375,000 | 0.21 |
| 406,062 | 210,000 | 0.48 |
| 282,766 | 410,000 | 0.45 |
| 486,445 | 205,000 | 0.58 |
| 486,895 | 375,000 | 0.23 |
| 430,277 | 285,000 | 0.34 |
| 229,629 | 210,000 | 0.09 |
| 182,805 | 455,000 | 1.49 |
| 380,680 | 585,000 | 0.54 |
| 396,166 | 375,000 | 0.05 |
| 252,980 | 330,000 | 0.30 |
| 289,373 | 285,000 | 0.02 |
| 210,614 | 330,000 | 0.57 |
| 308,166 | 210,000 | 0.32 |
| 534,333 | 530,000 | 0.01 |
| 416,904 | 535,000 | 0.28 |
| 203,400 | 410,000 | 1.02 |
| 220,600 | 330,000 | 0.50 |
| **Mape** | | **43.52** |

Dapat terlihat pada Table 6 Perbandingan Kinerja Algoritma K-Nearest Neighbor dan Naïve Bayes, pada penelitian ini algoritma K-Nearest Neighbor mempunyai kinerja yang lebih baik dibandingkan dengan algoritma Naïve Bayes dari segi waktu yang dibutuhkan dalam proses *training* maupun dari hasil *accuracy score*. Pada algoritma K-Nearest Neighbor didapatkan hasil *accuracy score* tertinggi dengan nilai 0,5714 pada variasi data 60:40, sedangkan untuk *accuracy score* tertinggi pada algoritma Naïve Bayes didapatkan oleh variasi data 20:80, 40:60, 50:50, dan 80:20 dengan nilai 0,4.

Pada penelitian ini, variasi data yang menghasilkan *accuracy score* tertinggi adalah variasi data *train* dan data *test* 60:40 dengan algoritma K-Nearest Neighbor.

Berdasarkan prediksi dari variasi data *train* dan data *test* tersebut, peneliti mengukur nilai *error rate absolute* dengan rumus MAPE untuk mengukur seberapa tepat atau akurat suatu prediksi yang dihasilkan[8] dan ditunjukkan pada Table 8 Hasil Perhitungan MAPE untuk Variasi Data 60:40 dengan Algoritma K-Nearest Neighbor. Untuk rumus MAPE, ditunjukkan pada persamaan 1 dan untuk kriteria MAPE akan ditunjukkan pada Table 7 Kriteria Nilai MAPE.

$$MAPE = \frac{100}{n}\sum_{1=0}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \quad (1)$$

Keterangan:

$\hat{y}_i = Hasil\,Prediksi$

$y_i = nilai\,aktual$

$n = banyaknya\,data\,yang\,diuji$

Pada perhitungan tersebut, didapatkan nilai MAPEnya sebesar 43,52, yang mana nilai tersebut masuk ke kategori "cukup" dalam memprediksi harga rumah pada penelitian ini.

## 4. Kesimpulan

Berdasarkan tujuan penelitian "Perbandingan Kinerja Algoritme Naïve Bayes dan K-Nearest Neighbor (KNN) Untuk Prediksi Harga Rumah", diperoleh kesimpulan bahwa model pembelajaran *lazy learning* memiliki kinerja yang lebih unggul dalam nilai *accuracy score* serta kecepatan dalam *training data* dengan nilai *accuracy score* dan waktu yang dibutuhkan adalah 0,5714 dan 0,0839 detik menggunakan algoritma K-Nearest Neighbor. Sedangkan *accuracy score* tertinggi yang didapatkan model pembelajaran *eager learning* adalah 0,4 dengan waktu *training data* tercepat selama 0,1615 detik dengan menggunakan algoritma Naïve Bayes.

Jika dilihat dari grafik waktu yang dibutuhkan pada proses *training data* dengan menggunakan algoritma K-Nearest Neighbor cenderung menurun, hal ini menunjukkan bahwa pernyataan metode pembelajaran *lazy learning* yang menghabiskan waktu lebih sedikit dalam fase *training* adalah benar[15]. Berdasarkan hasil pengujian yang dilakukan, algoritma K-Nearest Neighbor dan Naïve Bayes dapat dikatakan kedua algoritma tersebut kurang bisa memprediksi harga rumah dengan baik, dikarenakan nilai MAPE yang dihasilkan yaitu sebesar 43,52 hanya termasuk dalam kategori "cukup", bukan kategori "sangat baik". Melalui penelitian ini, peneliti tidak menyarankan hanya menggunakan penelitian ini sebagai acuan untuk melihat keunggulan dari metode pembelajaran *eager learning* dan *lazy learning*, karena jenis algoritma yang digunakan, *preprocessing* yang dilakukan, serta jenis *input* dan *output* yang diinginkan dapat mempengaruhi hasil prediksi dari pengujian yang dilakukan. Untuk mengetahui lebih jauh perihal kinerja metode pembelajaran mana yang lebih unggul, diperlukan untuk menambahkan variasi parameter pengujian pada penelitian.

## Referensi

[1]. Phan, T. D. (2019). Housing price prediction using machine learning algorithms: The case of Melbourne city, Australia. Proceedings - International Conference on Machine Learning and Data Engineering, ICMLDE 2018, 8–13.

[2]. Praningki, T., & Budi, I. (2018). Sistem Prediksi Penyakit Kanker Serviks Menggunakan CART, Naïve Bayes, dan k-NN. Creative Information Technology Journal, 4(2), 83.

[3]. Wahyono, T. (2018). Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan. *Gava Media*, *September 2018*, 49.

[4]. Varma, A., Sarma, A., Doshi, S., & Nair, R. (2018). House Price Prediction Using Machine Learning and Neural Networks. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, 1936–1939.

[5]. Ardyanti, Hesti; Goejantoro, Rito; Amijaya, F. D. T. (2019). *View of Perbandingan Metode Klasifikasi Naïve Bayes Dan Jaringan Saraf Tiruan.pdf*. Universitas Mulawarman. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5871/2799

[6]. Saputro, D. D., & Yulita, I. N. (2012). ANALISIS DAN IMPLEMENTASI ALGORITMA HYBRID ( EAGER LEARNING DAN LAZY LEARNING) PADA INTRUSION DETECTION SYSTEM. *Telkom University*.

[7]. Virkhansa, Chelsa Farah; Setiawan, Budi Darma; Dewi, C. (2019). *Tampilan Klasifikasi Status Gunung Berapi dengan Metode Learning Vector Quantization (LVQ).pdf*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.

[8]. Hudiyanti, C. V., Bachtiar, F. A., & Setiawan, B. D. (2019). Perbandingan Double Moving Average dan Double Exponential Smoothing untuk Peramalan Jumlah Kedatangan Wisatawan Mancanegara di Bandara Ngurah Rai. 3(3), 2667–2672.

[9]. Masykur, H. N. (2010). APLIKASI DATA MINING UNTUK MENAMPILKAN INFORMASI TINGKAT KELULUSAN MAHASISWA (Studi Kasus di Fakultas MIPA Universitas Diponegoro). Universitas Diponegoro.

[10]. Ambesange, S., Nadagoudar, R., Uppin, R., Patil, V., Patil, S., & Patil, S. (2020). Liver Diseases Prediction using KNN with Hyper Parameter Tuning Techniques. *Proceedings of B-HTC 2020 - 1st IEEE Bangalore Humanitarian Technology Conference*, 1–6.

[11]. Setiawan, Q. S., Rustam, Z., Hartini, S., Laeli, A. R., & Wirasati, I. (2020). Comparison of Naive Bayes and Decision Tree for Classifying Hepatocellular Carcinoma (HCC). *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies, 3ICT 2020*, 1–5. https://doi.org/10.1109/3ICT51146.2020.9312022

[12]. Towfek El-Kenawy, E.-S. M. (2019). *A Machine Learning Model for Hemoglobin Estimation and Anemia Classification Related papers Spam Det ect ion for Mobile Short Messaging Service Using Dat a Mining Classifiers A Machine Learning Model for Hemoglobin Estimation and Anemia Classification*.

[13]. Provost, F. (2000). Machine learning from imbalanced data sets 101. Proceedings of the AAAI'2000 Workshop on …, 3. https://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-001.pdf%5Cnpapers://1c40c143-2a6e-4e94-8c17-c5bc9ae73d7e/Paper/p11435

[14]. Wei, C. C. (2015). Comparing lazy and eager learning models for water level forecasting in river-reservoir basins of inundation regions. Environmental Modelling and Software, 63, 137–155.

[15]. Virkhansa, Chelsa Farah; Setiawan, Budi Darma; Dewi, C. (2019). Tampilan Klasifikasi Status Gunung Berapi dengan Metode Learning Vector Quantization (LVQ).pdf. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.