Auto Machine Learning dengan Menggunakan H2O AutoML untuk Prediksi Harga Bitcoin

Geadalfa Giyanda[1] dan Siti Saidah[2]

[1] Teknik Informatika, Fakultas Teknik Industri, Universitas Gunadarma
[2] Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
E-mail: geadalfa@student.gunadarma.ac.id, sitisaidah@staff.gunadarma.ac.id*)

Abstrak

Bitcoin merupakan salah satu penerapan konsep *cryptocurrency*, yang mengemukakan saran dan ide terhadap bentuk baru mata uang menggunakan kriptografi dengan fungsi untuk mengontrol pembuatan dan transaksi. Kesulitan memprediksi harga Bitcoin dapat terjadi, jika dilakukan secara manual, oleh karena itu dibutuhkan aplikasi untuk memprediksi harga Bitcoin dengan menggunakan bahasa pemrograman Python dinamis yang dilengkapi dengan manajemen memori otomatis. Python juga dapat berkolaborasi sebagai *library* dalam sebuah aplikasi salah satu contohnya adalah *library* H2O yang dibuat oleh H2O.ai. H2O telah memudahkan non-ahli untuk bereksperimen dengan *machine learning*, H2O AutoML dapat digunakan untuk mengotomatiskan alur kerja *machine learning*, yang mencakup pelatihan otomatis dan penyetelan banyak model dalam batas waktu yang ditentukan pengguna. Hasil evaluasi terhadap data uji dengan menggunakan model H2O AutoML dalam memprediksi harga pembukaan Bitcoin memperoleh nilai koefisien determinasi (R^2) sebesar 0.968 dan nilai *error* sebesar 3.48%.

Kata Kunci: Bitcoin, Machine Learning, Python, AutoML, H2O

Pendahuluan

Perekonomian Indonesia dipengaruhi oleh fluktuasi kurs mata uang asing. Banyak pihak yang memiliki kepentingan khusus untuk menyiapkan langkah strategis, agar tidak mengalami kerugian yang besar. Penelitian dengan judul Analisis *Support Vector Regression* (SVR), dalam Memprediksi Kurs Rupiah Terhadap Dollar Amerika Serikat mampu menghasilkan akurasi yang cukup baik pada kedua fungsi kernel yang digunakan, yaitu kernel linier dan kernel polynomial [1], penelitian yang mengamati Pemodelan Data Indeks Harga Saham Gabungan Menggunakan *Regresi Penalized Spline* menghasilkan nilai ketepatan model terbaik, dapat ditunjukkan dari besarnya nilai koefisien determinasi (R^2) dan MAPE [2].

Peneliti berikut memfokuskan Visualisasi Data untuk Memprediksi Pasar Saham dari Hasil Pengolahan Data Set S&P 500 dengan menggunakan bahasa pemrograman $R - Programming$ yang mampu menyimpulkan kemampuan untuk meningkatkan ketepatan dalam penyajian laporan transaksi [3]. Transaksi lain yang merupakan contoh yang proses transaksi dijalankan mirip dengan penjualan saham, yang memiliki fungsi yang sama sebagai alat tukar yaitu *Cryptocurrency*, berwujud mata uang digital atau mata uang virtual. *Cryptocurrency* yang digunakan menerapkan sistem kriptografi untuk mengamankan dan memverifikasi setiap transaksi, serta untuk mengontrol pembuatan unit-unit (token) baru dari suatu *cryptocurrency* tertentu. *Cryptocurrency* adalah entri yang terbatas dalam basis data yang tidak dapat diubah kecuali kondisi tertentu terpenuhi [4].

Bitcoin merupakan mata uang kripto yang telah paling terkenal sebagai mata uang digital. Jumlah Bitcoin beredar secara terbatas, sehingga Bitcoin baru dibuat dengan tingkat yang dapat diprediksi dan menurun, dengan perkataan lain permintaan harus mengikuti besaran inflasi untuk menjaga harga tetap stabil. Pasar Bitcoin relatif kecil dibanding potensi yang bisa dicapai, oleh sebab itu tidak dibutuhkan jumlah uang yang besar untuk pergerakan naik-turunnya harga di pasaran, dengan demikian harga bitcoin relatif mudah berubah [5].

Berlandaskan pengamatan di atas, peneliti tertarik untuk menerapkan konsep *Auto Machine Learning* dengan Menggunakan H2O Automl Untuk Prediksi Harga Bitcoin menggunakan Bahasa Pemrograman Python, didukung oleh *software*

DOI : http://dx.doi.org/10.32409/jikstik.20.2.2738,
*)Penulis korespondensi

Colaboratory oleh Google atau Jupyter Notebook yang bersifat *open source* dan bisa dijalankan pada semua komputer yang mempunyai sebuah web browser [6]. Batasan masalah penelitian ini adalah *Cryptocurrency* yang akan diprediksi adalah data historis Bitcoin dimulai dari 17 September 2014 – 26 Maret 2020 tergabung menjadi satu file csv terdiri dari 2018 baris dan 6 kolom.

File ekstensi program berbentuk .ipynb (*Interactive Python Notebook*) hanya bisa dijalankan dengan Google *Colaboratory* atau Jupter Notebook untuk menampilkan keluaran berupa grafik prediksi, koefisien determinasi (R^2) dan MAE (*Mean Absolute Error*) dijalankan terhadap data test yaitu 20% dari keseluruhan data. Penelitian ini bertujuan membuat aplikasi berekstensi .ipynb untuk memprediksi harga Bitcoin menggunakan Bahasa pemrograman Python dan *library* H2O AutoML dengan harapan mempemudah pengguna mengetahui pergerakkan harga Bitcoin.

Tinjauan Pustaka

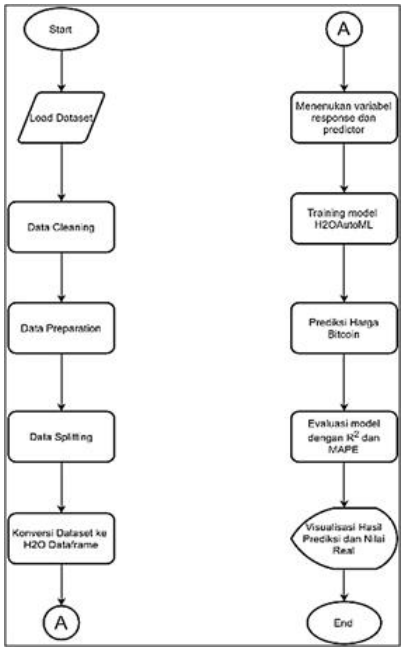
Sistem keamanan kripto yang ada pada mata uang digital atau mata uang virtual merupakan suatu keunggulan dibandingkan dengan mata uang konvensional, karena *Cryptography* menyediakan mekanisme digunakan untuk mengamankan sistem dalam mata uang digital dengan cara menyandikan atau mengkodekan aturan dalam sistem mata uang kripto itu sendiri [7]. Bitcoin yang diluncurkan pada tahun 2009, setelah melalui masa 8 tahun, tepatnya pada bulan Mei 2017 kapitalisasi pasar mata uang kripto aktif menembus batas $91 miliar.

Financial Stability Board (2018) menyebutkan bahwa pada tanggal 8 Januari 2018, kapitalisasi pasar gabungan aset kripto naik hingga $830 miliar, di mana sekitar 35% disebabkan oleh Bitcoin. Kondisi Bitcoin tetap mendominasi pasar, namun di sisi lain Bitcoin dihadapi oleh masalah teknis dan juga peningkatan teknologi mata uang kripto yang lain [4].

Python merupakan bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat kebacaan kode [8]. Situs research.google.com Google Colaboratory atau "Google Colab" adalah produk dari Google Research yang memungkinkan siapa saja untuk dapat menulis dan mengeksekusi kode python melalui browser, juga sangat cocok untuk diterapkan pada pembelajaran mesin (*Machine Learning*), analisis data, dan pendidikan.

Secara teknis, Colab adalah layanan Jupyter Notebook yang dihosting yang tidak membutuhkan pengaturan untuk digunakan, fasilitas lain yang tersedia antara lain akses gratis ke sumber daya komputasi termasuk *Grapichal Processing Unit* (GPU). Google Colab memiliki beberapa keuntungan yang bisa dimanfaatkan secara gratis, diantaranya adalah: *Graphical Processing Unit* (GPU), *Colaborate*, mudah terintegrasi dan fleksibel.

H2O.ai adalah sebuah perusahaan yang bergerak dibidang *Artificial Intelligence* dan *Machine Learning* yang menyediakan berbagai *Library Open Source* yang bernama H2O juga dengan maksud untuk digunakan secara bebas untuk semua orang. H2O AutoML dapat digunakan untuk mengotomatiskan alur kerja *machine learning*, yang mencakup pelatihan otomatis dan penyetelan banyak model dalam batas waktu yang ditentukan pengguna. Kumpulan model *stacked* atau sebuah model yang terbentuk dari tumpukkan beberapa model yang dilatih sebelumnya, lalu pada model terbaik dari masing-masing jenis akan secara otomatis dilatih pada koleksi masing-masing model untuk menghasilkan model ansambel yang sangat prediktif, dalam banyak kasus menjadi model dengan kinerja terbaik di AutoML Leaderboard [9].



Gambar 1: Aplikasi *Auto Machine Learning*

Metode Penelitian

Analisis untuk kebutuhan spesifikasi perangkat keras (*hardware*) terdiri dari Personal Computer, CPU AMD Ryzen 5 3600 6C/12T 3.6Ghz, RAM : 16GB DDR4, Hardisk : 2TB + 120GB SSD, GPU : Nvidia GTX 1070Ti, sedangkan spesifikasi perangkat lunak (*software*) yang digunakan adalah Python 3.8 dan Google Colab. Aplikasi dibuat dengan menampilkan data harga Bitcoin mentah dimaksudkan harga yang ditampilkan belum diolah, untuk selanjutnya diubah menjadi tabel, sehingga bisa divisualisasikan dengan baik menggunakan grafik. Agar pengguna dapat melihat harga Bitcoin yang fluktuatif dengan mudah, proses

berikutnya data dipisahkan menjadi data *train* dan data *test* dengan tujuan dapat dilakukan pembuatan model dan proses training terhadap model dengan menggunakan library H2O AutoML, model yang terbentuk diprediksi terhadap data test dan visualisasi ditampilkan dengan grafik.

Tahapan penelitian meliputi tahap data *input*, data *preprocessing* meliputi proses data *cleaning*, data *preparation* dan data *splitting*, lalu melakukan konversi data *train* dan data *test* ke H2O DataFrame, dilanjutkan ke proses training dan testing, menghitung Koefisien Determinasi (R^2) dan *Mean Absolute Percentage Error* (MAPE) terhadap tingkat kesalahan hasil prediksi, rancangan tampilan visualisasi dan tahap akhir dilakukan visualisasi hasil prediksi terhadap nilai aktual berdasarkan periode yang diujikan terhadap model prediksi. *Flowchart* pada Gambar 1 dapat menjelaskan proses-proses yang ada pada penelitian ini. *Flowchart* pada gambar 1 dijelaskan pada tahapan rincian berikut ini:

1. Tahap Load Dataset

Proses "*Load Dataset*" merupakan tahap data *input* yang mengolah Data Bitcoin harian pada periode 17 September 2014 sampai dengan 26 Maret 2020, data tersebut diperoleh dari situs Yahoo! Finance. Data Bitcoin tersebut terdiri dari beberapa kolom yaitu : *Date*, *Open*, *High*, *Low*, seperti pada Tabel 1 sebagai dari hasil dari input data menjadi Pandas DataFrame.

Tabel 1: Harga Bitcoin Dengan Pandas DataFrame

Date	Open	High	Low	Close	Adj Close	Volume
2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800.0
2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200.0
2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700.0
2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600.0
2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100.0

Penjelasan Tabel 1 yang dimaksud dengan *Open* adalah harga pembukaan dalam bursa perdagangan Bitcoin. Pada tanggal 17 September 2014, harga pembukaan Bitcoin adalah 465.864014 dalam satuan Dolar Amerika Serikat ("US$") yang artinya harga 1 Bitcoin adalah $465.864.

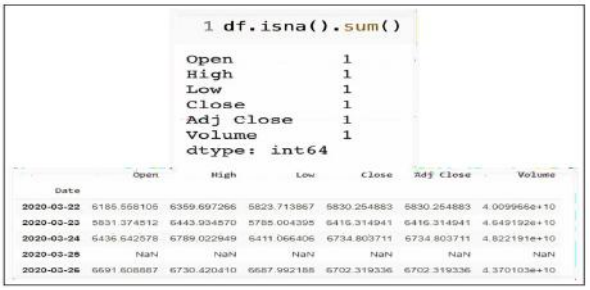
High adalah harga tertinggi pada saat bursa perdagangan, *Low* adalah harga terendah dalam bursa perdaganan, *Close* adalah harga saat penutupan bursa perdagangan, *Adj Close* adalah harga penutupan yang disesuaikan dengan aksi korporasi seperti *right issue*, *stock split* atau *stock reverse* dan Volume adalah jumlah transaksi jual atau beli yang dilakukan selama bursa perdagangan buka.

2. Tahap Data Preprocessing

Tahap data *preprocessing* dilakukan untuk membentuk data Bitcoin yang masih "mentah" menjadi bentuk data yang dapat diterima dan dipelajari polanya (*pattern*) oleh model yang akan dibuat. Tahap ini terdiri dari tiga proses yaitu data *cleaning*, data *preparation* dan data *splitting*. Hasil yang diperoleh adalah data *input* yang terbagi menjadi data *train* dan *test*, untuk selanjutnya akan dikonversi ke H2O DataFrame dan siap digunakan untuk data latih, pengujian model dan visualisasi hasil prediksi.

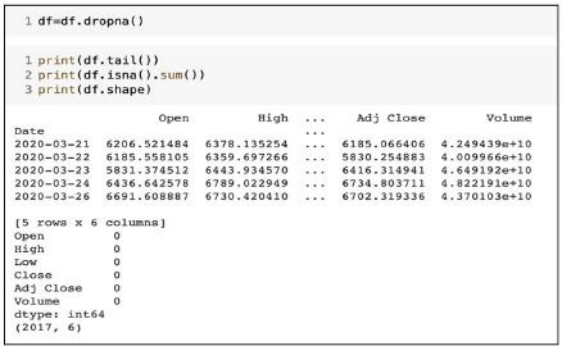
2.1 Data Cleaning

Data Bitcoin yang diperoleh dari situs Yahoo! Finance mempunyai beberapa sampel yang null atau "NaN", hal ini dikarenakan setiap tahun terdapat hari libur bursa perdagangan *cryptocurrency* yang mengikuti tanggal hari libur di Amerika. Data *cleaning* untuk membersihkan sampel tersebut dilakukan, karena model prediksi tidak bisa menerima masukkan yang tidak mempunyai nilai atau "NaN".



Gambar 2: Sampel Data yang Mengandung NaN

Data *cleaning* berfungsi untuk membersihkan sampel tersebut karena model prediksi yang akan dibuat tidak bisa menerima masukkan yang tidak mempunyai nilai atau "NaN", pada Gambar 2 menggunakan fungsi dari library Pandas untuk melihat jumlah nilai NaN pada sampel. Fungsi ".isna()" untuk mencari nilai NaN dan fungsi ".sum()" untuk menjumlahkan berapa sampel yang mengandung nilai NaN tersebut.



Gambar 3: Contoh (*sample*) yang Telah Dibersihkan

Cara menghapus contoh (*sample*) dengan menggunakan fungsi ".dropna()" yang telah tersedia dari library Pandas lalu melihat kembali apakah masih ada nilai contoh (*sample*) yang "NaN" atau tidak dengan menggunakan tiga fungsi "print" yang di dalamnya terdapat beberapa fungsi dari library Pandas yaitu fungsi "df.tail()" untuk melihat 5 contoh (*sample*), "df.isna().sum()" untuk melihat dan menjumlahkan nilai NaN yang ada pada dataset yang telah dimasukkan ke dalam variabel df dan "df.shape" dan terakhir untuk melihat berapa baris dan kolom yang ada pada dataset, selanjutnya dapat diketahui bahwa data Bitcoin sudah tidak memiliki nilai NaN dan memiliki 2017 contoh (*sample*)/baris. Langkah proses penghapusan dapat ditampilkan pada gambar 3. Tampilan pada Gambar 3 menjelaskan bahwa kolom *open*, *high*, *low*, *close*, *adj close* dan volume menunjukkan angka *null*.

2.2. Data Preparation

Dataset Bitcoin yang telah dibersihkan, dilanjutkan ke tahap data *preparation* untuk mendapatkan model distribusi *lag*. Analisis regresi yang melibatkan data runtun waktu, jika model regresi memasukan tidak hanya nilai variabel bebas saat ini atau X_1 tetapi juga nilai variabel bebas masa lalu pada waktu $t-1$, $t-2$ dan seterusnya [10]. Model regresi seperti ini disebut model distribusi *lag*, ketika distribusi *lag* telah dilakukan pada dataset Bitcoin, maka kolom pada dataset tersebut bertambah menjadi 34 kolom. Model regresi seperti ini disebut model *distributed lag*. Berikut ditunjukkan rumus dari distribusi *lag* [14].

$$Y_t = \alpha + \beta_0 X_{t-1} + \beta_1 X_{t-1} + \beta_2 X_{t-2} + u_t$$

Rumus distribusi *lag* yang telah diproses pada dataset Bitcoin, maka kolom pada dataset tersebut bertambah menjadi 34 kolom. Berikut ditunjukkan pada Gambar 4 hasil kolom yang terbentuk oleh karena distribusi *lag* dan pada Gambar 4 adalah beberapa kolom harga pembukaan ("*Open*") yang terbuat dari distribusi lag beserta *value* di dalamnya.

```
Index(['Open', 'Open_lag1', 'Open_avg_window_length2', 'Open_lag2',
       'Open_avg_window_length3', 'Open_lag3', 'Open_avg_window_length4',
       'High_lag1', 'High_avg_window_length2', 'High_lag2',
       'High_avg_window_length3', 'High_lag3', 'High_avg_window_length4',
       'Low_lag1', 'Low_avg_window_length2', 'Low_lag2',
       'Low_avg_window_length3', 'Low_lag3', 'Low_avg_window_length4',
       'Close_lag1', 'Close_avg_window_length2', 'Close_lag2',
       'Close_avg_window_length3', 'Close_lag3', 'Close_avg_window_length4',
       'Adj Close_lag1', 'Adj Close_avg_window_length2', 'Adj Close_lag2',
       'Adj Close_avg_window_length3', 'Adj Close_lag3',
       'Adj Close_avg_window_length4', 'Volume_lag1', 'Volume_lag2',
       'Volume_lag3'],
      dtype='object')
```

Gambar 4: Seluruh Index Kolom Setelah Lag Terdistribusi

Gambar 4 menjelaskan bahwa semua index yang ada, kecuali "*Open*" akan dipakai sebagai variabel *X* untuk proses training pada model dan variabel – variabel tersebut tidak perlu untuk diolah, karena akan diolah secara otomatis dengan menggunakan model H2O AutoML. Harga pembukaan setelah dilakukannya distribusi *lag* terlihat pada Gambar 5, distribusi lag membuat sebuah kolom harga pembukaan menjadi 7 kolom harga pembukaan.

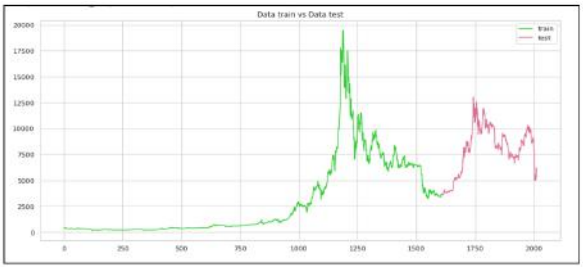
Date	Open	Open_lag1	Open_avg_window_length2	Open_lag2	Open_avg_window_length3	Open_lag3	Open_avg_window_length4
2014-09-17	465.864014	456.859985	440.481491	424.102997	428.211995	394.673004	420.930244
2014-09-18	456.859985	424.102997	409.388000	394.673004	408.963664	408.084991	406.450249
2014-09-19	424.102997	394.673004	401.378997	408.084991	400.619334	399.100006	400.987503
2014-09-20	394.673004	408.084991	403.592499	399.100006	403.092306	402.092010	411.257003
2014-09-21	408.084991	399.100006	400.596008	402.092010	412.314341	435.751007	415.024757
2014-09-22	399.100006	402.092010	418.921509	435.751007	420.333008	423.156006	418.107502
2014-09-23	402.092010	435.751007	429.453507	423.156006	423.445333	411.428986	418.473000
2014-09-24	435.751007	423.156006	417.292495	411.428986	412.719564	403.556000	409.403000
2014-09-25	423.156006	411.428986	407.492493	403.556000	404.818665	399.471008	397.846001
2014-09-26	411.428986	403.556000	401.513504	399.471008	393.318339	376.928009	389.010757

Gambar 5: Kolom Harga Pembukaan

Gambar 5 menjelaskan secara singkat perbedaan antara kolom harga *Open* yang asli dengan kolom yang lain, dengan memperhatikan *lag* terdistribusi, maka terjadi pergeseran nilai pada kolom "Open_lag1" adalah nilai yang ada pada baris kedua pada kolom "*Open*", begitu juga nilai yang ada pada "Open_lag2" adalah nilai dari baris kedua pada kolom "Open_lag1" begitu pula sampai kepada "Open_lag3". Lalu, nilai yang ada di dalam kolom "Open_avg_window_length2" adalah rata-rata dari hasil penjumlahan kolom "*Open*" dan "Open_lag1" per baris, begitu juga dengan kolom "Open_avg_window_length" dan seterusnya .

2.3. Data Splitting

Dataset Bitcoin selanjutnya dibagi menjadi data *train* dan data *test*, seperti visualiasi yang ada pada Gambar 6 di bawah ini. Pembagian data ini berguna untuk menentukan data yang ingin dilatih dan data yang akan menjadi evaluasi model untuk menentukan ke akuratan prediksi dan tingkat kesalahan (*error*).



Gambar 6: Visualisasi Pembagian Data Train dan Data Test

Gambar 6 menunjukan Pembagian dilakukan dengan presentasi data *train* 80% dan data *test* 20%, sehingga diperoleh *dataset* Bitcoin dataset *train* sebanyak 1611 sampel/baris dan 34 kolom, sedangkan dataset *test* memiliki 403 sampel/baris dan 34 kolom.

3. Tahap Konversi DataFrame

Data *train* dan data *test* berada pada format Pandas DataFrame, namun untuk menggunakan model H2O AutoML Pandas DataFrame tidak bisa digunakan sebagai *input*. H2O mewajibkan penggunaan H2O DataFrame sebagai *input* agar bisa menggunakan model H2O AutoML. Berikut proses penguraian dan mengonversikan data *train* dan data *test* dari Pandas DataFrame ke H2O DataFrame.

Gambar 7: Parse Pandas DataFrame ke H2O DataFrame

Gambar 7 menunjukkan proses parse atau penguraian dan mengonversikan data *train* dan data *test* dari Pandas DataFrame ke H2O DataFrame membutuhkan waktu yang cukup singkat (kisaran waktu yang dibutuhkan kurang dari satu menit). Hasil dari proses mengonversikan Pandas DataFrame ke H2O DataFrame, H2O DataFrame hampir tidak berbeda dengan Pandas DataFrame tetapi terlihat jelas bahwa H2O DataFrame tidak memiliki garis pembatas antar sel dan tidak memiliki kolom indeks yang biasa terletak paling kiri yang memiliki indeks baris berbentuk tanggal.

4. Permodelan dan Prediksi

Langkah berikutnya adalah konversikan *dataset* Bitcoin ke dalam H2O DataFrame, maka yang harus dilakukan selanjutnya adalah pembuatan model H2O AutoML hingga memprediksikan harga pembukaan Bitcoin dengan menggunakan model yang telah dibuat. Tetapi sebelum melanjutkan untuk membuat model hingga proses prediksi, ada beberapa tahapan yang harus dilalui diantaranya menentukan variabel X ("*Predictors*") dan$y$ ("*Response*"), membuat model H2O AutoML, melatih ("*training*") model, memilih model terbaik dan memprediksi harga pembukaan ("*Open*") Bitcoin dengan menggunakan model terbaik.

4.1. Menentukan Variabel Predictor dan Response

Semua model H2O mengharuskan untuk memasukkan variabel X ("*Predictors*") dan Y ("*Response*"). Variabel *predictor* adalah sebuah variabel masukkan atau *input* yang bersifat independen/bebas biasanya dilambangkan dengan huruf 'X' sedangkan response adalah variabel keluaran atau *output* yang bersifat dependen/bergantung kepada variabel masukkannya, biasanya variabel response dilambangkan dengan huruf 'Y'. Gambar 8 menunjukkan proses pembentukan variabel X dan variabel Y.

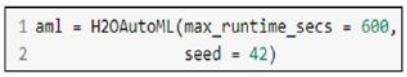
Gambar 8 menjelaskan bahwa penggunaan model H2O AutoML tidak perlu sulit untuk menentukan variabel X dan Y, karena dengan *auto machine learning*, model tersebut yang akan menentukkan dan mengolah variabel *predictor* itu sendiri, oleh sebab itu pengguna bisa memasukkan semua kolom ke dalam variabel *predictor* kecuali kolom yang ingin diprediksi, dalam kasus ini kolom "*Open*" akan menjadi variabel response atau variabel 'y' sehingga kolom "*Open*" tidak akan dimasukkan ke dalam variabel X atau predictor.

```
[43]  1 x

['Open_lag1',
 'Open_avg_window_length2',
 'Open_lag2',
 'Open_avg_window_length3',
 'Open_lag3',
 'Open_avg_window_length4',
 'High_lag1',
 'High_avg_window_length2',
 'High_lag2',
 'High_avg_window_length3',
 'High_lag3',
 'High_avg_window_length4',
 'Low_lag1',
 'Low_avg_window_length2',
 'Low_lag2',
 'Low_avg_window_length3',
 'Low_lag3',
 'Low_avg_window_length4',
 'Close_lag1',
 'Close_avg_window_length2',
 'Close_lag2',
 'Close_avg_window_length3',
 'Close_lag3',
 'Close_avg_window_length4',
 'Adj Close_lag1',
 'Adj Close_avg_window_length2',
 'Adj Close_lag2',
 'Adj Close_avg_window_length3',
 'Adj Close_lag3',
 'Adj Close_avg_window_length4',
 'Volume_lag1',
 'Volume_lag2',
 'Volume_lag3']

[44]  1 y

'Open'
```

Gambar 8: Gambar 8. Variabel X dan Y

4.2. H2O AutoML Model

Variabel masukkan dan variabel keluaran yang telah didefinisikan, merupakan keseluruhan syarat yang harus dipenuhi untuk membuat model H2O AutoML. Proses pendefinsian Model H2O AutoML dapat dilihat pada Gambar 9.



```
1 aml = H2OAutoML(max_runtime_secs = 600,
2                 seed = 42)
```

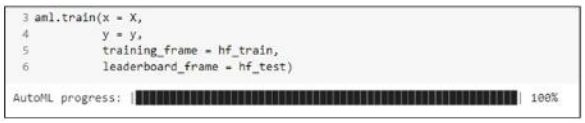
Gambar 9: Mendefinisikan Model H2O AutoML

Kemudahan membuat model H2O AutoML dapat dijelaskan pada Gambar 9, pengguna hanya memasukkan fungsi "H2OAutoML()" ke dalam sebuah variabel yang bernama "aml", di dalam fungsi model tersebut terdapat parameter yang bisa dikosongkan karena sudah menjadi fitur sebuah model *auto machine learning* akan berjalan secara otomatis, namun peneliti memasukkan dua parameter yaitu parameter "max_runtime_secs" untuk

mengatur waktu *running* model dan "*seed*" untuk memastikan bahwa angka acak yang dihasilkan oleh algoritma *machine learning* selalu sama setiap kali dijalankan.

4.3. Proses Training Model H2O AutoML

Tahap selanjutnya adalah proses melatih atau training model H2O AutoML. Tahap melatih model adalah tahap yang paling lama untuk dijalankan, tetapi dapat diatasi dengan menggunakan parameter "max_runtime_secs" pengguna bisa merubah waktu maksimal yang digunakan untuk melatih model di dalam model H2O AutoML, yang dapat dijelaskan pada Gambar 10 berikut ini :

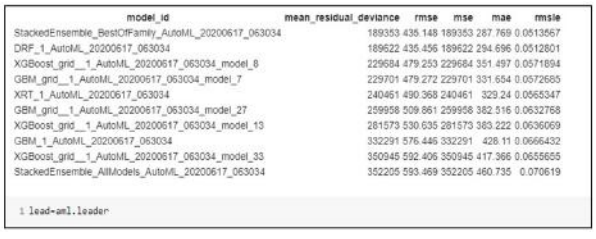


Gambar 10: Proses Training Model H2O AutoML

Gambar 10 menjelaskan bahwa variabel "aml" yang di dalamnya telah dimasukkan model H2O AutoML dilatih dengan menggunakan fungsi "*train*" yang di dalamnya terdapat empat buah parameter yaitu parameter variabel masukkan (X), parameter variabel keluaran (Y), DataFrame untuk melatih model (training_frame) dan DataFrame untuk memvalidasi model (leaderboard_frame). Proses pelatihan model berjalan hingga maksimal 600 detik atau 10 menit karena telah dideklarasikan saat pembuatan model.

4.4. Pemilihan Model Terbaik

Proses pelatihan model H2O AutoML selama sepuluh menit akan menghasilkan 10 model terbaik dengan berbagai macam algoritma dan pengaturan dari setiap model yang disesuaikan untuk menghasilkan tingkat kesalahan ("*Error*") yang paling kecil, sehingga pengguna tidak perlu mengkonfigurasi model satu per satu untuk mendapatkan model terbaik dari dataset saat ini, Gambar 11 menunjukkan proses pemilihan 10 model tebaik.



model_id	mean_residual_deviance	rmse	mse	mae	rmsle
StackedEnsemble_BestOfFamily_AutoML_20200617_063034	189353.435.148	189353	287.769	0.0513567	
DRF_1_AutoML_20200617_063034	189622.435.456	189622	294.696	0.0512801	
XGBoost_grid__1_AutoML_20200617_063034_model_8	229684.479.253	229684	351.497	0.0571894	
GBM_grid__1_AutoML_20200617_063034_model_7	229701.479.272	229701	331.654	0.0572685	
XRT_1_AutoML_20200617_063034	240461.490.368	240461	329.24	0.0565347	
GBM_grid__1_AutoML_20200617_063034_model_27	259958.509.861	259958	382.516	0.0632768	
XGBoost_grid__1_AutoML_20200617_063034_model_13	281573.530.635	281573	383.222	0.0636069	
GBM_1_AutoML_20200617_063034	332291.576.445	332291	428.11	0.0666432	
XGBoost_grid__1_AutoML_20200617_063034_model_33	350945.592.406	350945	417.366	0.0655655	
StackedEnsemble_AllModels_AutoML_20200617_063034	352205.593.469	352205	460.735	0.070619	

1 lead-aml.leader

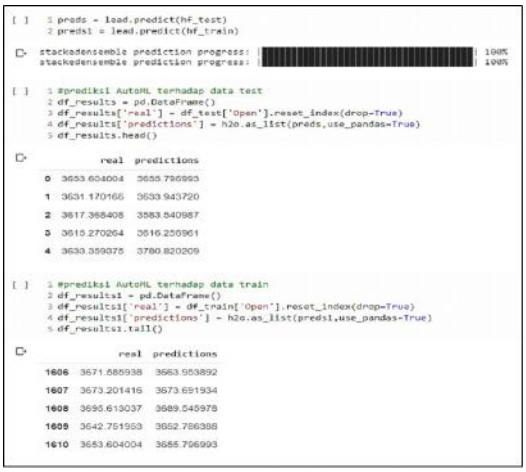
Gambar 11: Sepuluh Model Terbaik

Model terbaik yang dihasilkan diurut pada Gambar 11, dapat dikelompokkan berdasarkan 5 macam kesalahan diantaranya adalah Mean_Residual_Deviance, RMSE ("*Root Mean Squared Error*"), MSE ("*Mean Squared Error*"), MAE ("*Mean Absolute Error*") dan RMSLE ("*Root Mean Squared Logarithmic Error*"), semua kesalahan ini berguna untuk mengevaluasi model yang telah menggunakan dataset Bitcoin. Sepuluh model terbaik tersebut memiliki model_id yang sama contohnya seperti StackedEnsemble, XGBoost dan GBM. Namun, di dalam setiap model terdapat konfigurasi yang berbeda-beda yang sudah teroptimasi secara otomatis, hal itu bisa dilihat dari nama belakangnya yang menggunakan angka atau keterangan seperti model_8 pada model XGBoost diurutan ke 3 model terbaik.

Pengamatan pada sepuluh model terbaik yang dibuat secara otomatis oleh AutoML, maka sudah diputuskan bahwa model H2O AutoML dengan model_id "StackedEnsemble_BestOfFamily_AutoM L_20200617_063034" sebagai model terbaik, karena memiliki tingkat kesalahan terkecil pada 5 macam kesalahan ("*Error*"). Model teratas dari sepuluh model terbaik ini lalu dimasukkan ke dalam variabel "*lead*" secara otomatis dengan menggunakan fungsi "*leader*" yang telah ada secara default pada library H2O AutoML.

4.5. Prediksi Harga Pembukaan Bitcoin

Model terbaik H2O AutoML yang telah dimasukkan ke dalam variabel "lead" lalu digunakan untuk memprediksi harga pembukaan Bitcoin dari data *train* dan data *test*. Tujuan memprediksi dari kedua dataset ini untuk memastikan bahwa tingkat error model lebih besar terhadap data *test* dan lebih kecil terhadap data *train*, dapat dijelaskan pada Gambar 12 berikut ini.



Gambar 12: Memprediksi Harga Pembukaan Bitcoin

Gambar 12 menerangkan bahwa variabel "preds" dan "preds1" digunakan untuk menyimpan hasil dari prediksi model H2O AutoML terbaik, variabel "preds" untuk menyimpan hasil prediksi dari data test dan "preds1" menyimpan hasil prediksi dari data train. Hasil prediksi tersebut kemudian diubah menjadi variabel "df_results" untuk data test

dan "df_results1" untuk data train dengan menggunakan format Pandas DataFrame. Lalu peneliti mencoba untuk menampilkan masing-masing 5 data nilai asli ("real") dan hasil prediksinya terhadap kedua dataset tersebut.

5. Evaluasi Model

Evaluasi dibutuhkan untuk mengetahui keakuratan sebuah model *machine learning*, karena dari evaluasi diperoleh hasil perhitungan dari akurasi, performa dan tingkat kesalahan ("*error*"). Hasil prediksi dari model H2O AutoML bisa dievaluasi ke akuratannya dengan menggunakan "Koefisien Determinasi" yang dilambangkan dengan *R Squared* (R^2) dan tingkat *error* model dapat dievaluasi dengan menggunakan metode *Mean Absolute Percentage Error* (MAPE) karena metode ini sangatlah sesuai untuk masalah prediksi statistik yang ada pada dataset Bitcoin.

Perhitungan kemampuan sejumlah variabel bebas yang ada dalam model persamaan regresi linier berganda dapat diperoleh dari Koefisien Determinasi (R^2) secara bersamaan dan mampu menjelaskan variabel tidak bebasnya. Nilai R^2 berada di rentang 0 sampai 1. Perolehan Nilai di atas angka 0,5 dapat dikategorikan 'baik', sebaliknya nilai R^2 di bawah 0,5, dapat diketegorikan sebagai nilai 'tidak baik'.

Umumnya acuaan yang digunakan dari hasil penghitungan koefisien determinasinya, maka sebuah model regresi yang dihasilkan dari H2O AutoML bisa dikatakan "baik" untuk digunakan apabila nilai R^2 di atas angka 0,5. Hal ini karena sebagian besar variabel terikatnya mampu dijelaskan dengan baik oleh variabel bebasnya. Sebaliknya, model regresi linier dianggap "tidak baik"digunakan apabila nilai R^2 di bawah 0,5. Rumus koefisien determinasi yang digunakan sebagai berikut:

$$R^2 = \frac{SSR}{SST}$$

Keterangan:
R2 = Koefisien Determinasi
SSR = *Regression Sum of Squares*
SST = Total *Sum of Squares*

MeanAbsolute Percentage Error (MAPE) atau rata-rata persentase kesalahan absolut, juga dikenal sebagai rata-rata persentase absolut, adalah ukuran akurasi prediksi metode peramalan dalam statistik, misalnya dalam estimasi tren, juga digunakan sebagai fungsi kerugian ("*loss function*") untuk masalah regresi dalam *machine learning*. Rumus dari MAPE adalah sebagai berikut :

$$MAPE = \frac{1}{n}\sum \left| \frac{Actual - Forecast}{Actual} \right| X100$$

Rumus di atas, jika *Actual* sama dengan *Forecast*, maka MAPE berharga nol, yang artinya semakin nilai MAPE mendekati 0 maka model sangat akurat dan tidak ada kesalahan dalam memprediksi. Penelitian ini menghasilkan Nilai Akurasi R^2 dan error sebagai berikut :

1. Akurasi R^2 dan *Error* prediksi vs *test*
   R^2 = 0.96811895122219353
   $MAPE$= 3.48%
2. Akurasi R^2 dan *Error* prediksi vs *train*
   R^2 = 0.9992115633449767
   $MAPE$= 1.38%

Hasil dari evaluasi model telah didapatkan untuk kedua dataset yaitu data *train* dan data *test* adalah nilai koefisien determinasi (R^2) sebesar 0.968 untuk data uji, 0.999 untuk data latih dan nilai *MAPE* sebesar 3.48% untuk data uji dan 1.38% untuk data latih.

6. Perancangan Tampilan Program

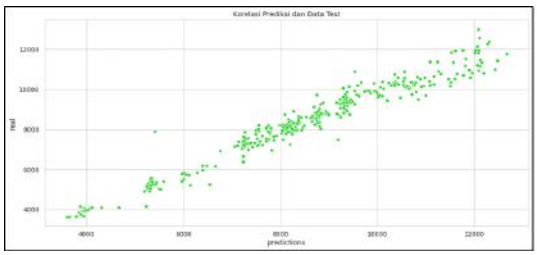
Nilai akurasi dan *error* tidak dapat mudah dipahami oleh sebagian orang, maka visualisasi dalam bentuk grafik pada program sangat diperlukan untuk mengatasi masalah tersebut. Membuat tampilan program bisa dilakukan dengan beberapa cara salah satunya bisa menggunakan koding dengan bahasa pemrograman Python, karena Python memiliki beberapa *library* yang dibuat khusus untuk menampilkan grafik dari data asli dan data yang telah diprediksi oleh model H2O AutoML.

Tampilan program yang digunakan adalah grafik plot korelasi dan grafik plot garis, karena dataset yang berbentuk tabel dan mempunyai sifat *time series* sangat cocok dan valid apabila ditampilkan dengan menggunakan kedua plot tersebut. Solusi yang diberikan ada pada list perintah program berikut ini:

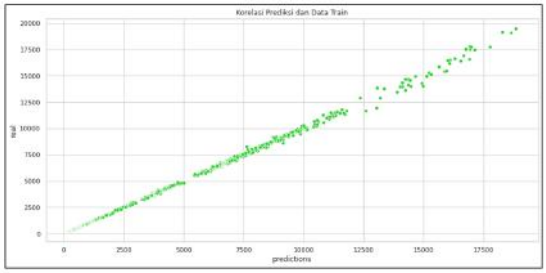
```
plt.figure(figsize=(15,7))
sns.set(style="whitegrid")
plt.title('Korelasi Prediksi dan Data Test')
sns.scatterplot(data=df_results, x=df_results['predictions'],y=df_results['real'],color='limegreen')

plt.figure(figsize=(15,7))
sns.set(style="whitegrid")
plt.title('Korelasi Prediksi dan Data Train')
sns.scatterplot(data=df_results1, x=df_results1['predictions'],y=df_results1['real'],color='limegreen')
```

Langkah pengkodean dengan menggunakan bahasa pemrograman Python yaitu menentukan panjang dan lebar plot visualisasi seperti yang terlihat pada list program di atas, baris pertama dalam kode digunakan agar plot visualisasi memiliki panjang 15 dan lebar 7, setelah itu menentukan warna latar pada plot visualisasi menjadi warna putih, lalu menuliskan judul pada plot, sehingga pengguna dapat melihat plot yang memiliki warna yang berbeda. Tahap akhir adalah menentukan jenis plot visualisasi dan data yang ingin ditampilkan serta warna dari plot itu sendiri, pada kasus ini plot yang akan digunakan adalah *scatter* plot atau diagram pencar yaitu plot visualisasi dengan bentuk

titik-titik yang berfungsi untuk melakukan pengujian terhadap seberapa kuatnya hubungan antara 2 (dua) variabel serta menentukan jenis hubungan dari 2 (dua) variabel tersebut apakah hubungan positif, hubungan negatif ataupun tidak ada hubungan sama sekali. Data yang digunakan adalah dataset asli dari data latih dan data uji serta data prediksi untuk tiap-tiap dataset yang akan terbagi menjadi dua plot.



Gambar 13: Tampilan Output dari Pengkodean Plot Korelasi



Gambar 14: Tampilan Output dari Pengkodean Plot Korelasi

Gambar 13 dan 14 adalah tampilan *output* dari pengkodean yang telah dilakukan, terlihat dua plot yang berbeda tingkat kerapatannya yang disebabkan oleh dataset yang berbeda sebagai input dari data. Namun terlihat juga kedua plot ini memiliki hubungan yang positif. Pengkodean Tampilan Plot Garis dapat diperoleh dengan menjalankan baris perintah berikut ini:

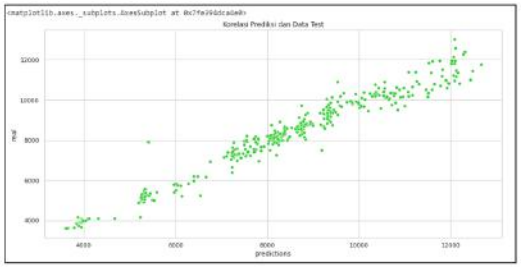
```
plt.figure(figsize=(15,7))
sns.set(style="whitegrid")
plt.title('Harga Open Bitcoin Data Test vs Prediksi')
sns.lineplot(data=df_results['real'], label='real',color='limegreen')
sns.lineplot(data=df_results['predictions'], label='prediksi',color='pale
violetred')

plt.figure(figsize=(15,7))
sns.set(style="whitegrid")
plt.title('Harga Open Bitcoin Data Train vs Prediksi')
sns.lineplot(data=df_resultsl['real'], label='real',color='limegreen')
sns.lineplot(data=df_resultsl['predictions'], label='prediksi',color='pal
evioletred')
```

Langkah pengkodean untuk menampilkan plot visualisasi garis tidak jauh berbeda dengan langkah untuk membuat plot visualisasi korelasi. Pengkodean untuk tampilan plot garis hanya berbeda pada jenis plotnya yaitu *line plot*, lalu untuk membandingkan dengan kedua data prediksi dan kedua dataset, maka dalam setiap *plot* harus dibuat 2 *line plot* dengan *dataset*, label dan warna yang berbeda agar pembaca atau orang yang melihatnya dapat membedakan dan membandingkan keakuratan data prediksi terhadap data aslinya.

Hasil dari pengkodean tersebut menunjukkan dua buah diagram garis yang di dalamnya terdapat nilai prediksi terhadap data latih dan data uji. Plot garis tersebut menunjukkan garis hijau untuk data prediksi dan warna merah untuk nilai aslinya yang diambil dari data latih maupun data uji.

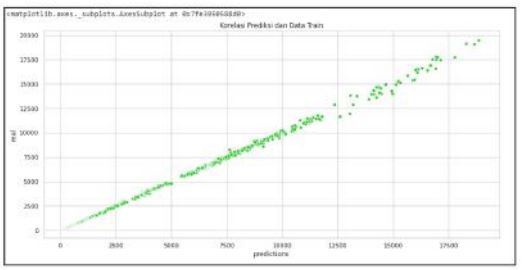
7. Visualisasi Hasil Prediksi

Visualisasi hasil prediksi model H2O AutoML merupakan *output* atau keluaran dari program yang peneliti buat agar memudahkan siapapun yang melihatnya. Pengkodean yang telah dibuat mempunyai *output* yang berbentuk diagram pencar atau disebut juga dengan *scatter plot* ada pada Gambar 15.



Gambar 15: Plot Korelasi Data Prediksi Terhadap Data Uji

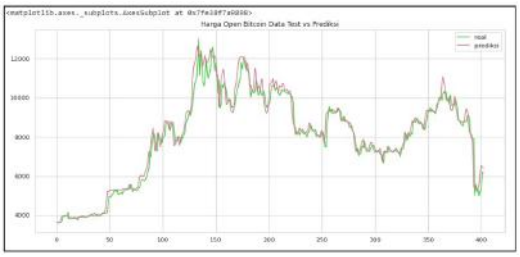
Plot korelasi pada Gambar 15 mempunyai sumbu x yang memiliki nilai data prediksi dan sumbu y yang memiliki nilai asli dari data uji menunjukkan bahwa pesebaran titik-titik atau dot-dot yang ada di dalam plot dipengaruhi oleh hasil prediksi terhadap data uji. Apabila hasil prediksi tidak sama bahkan jauh dari nilai asli data uji maka titik-titik tersebut akan menyebar ke berbagai tempat dan tidak bisa membentuk sebuah garis lurus yang rapat, hal ini disebabkan karena model memprediksi nilai atau value yang tidak ada pada saat proses training.



Gambar 16: Plot Korelasi Data Prediksi Terhadap Data Latih

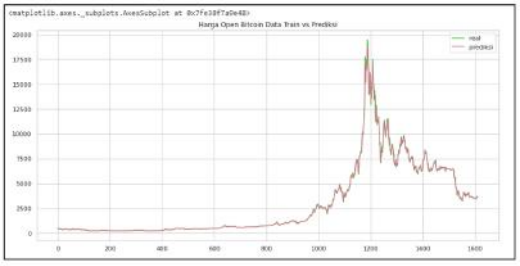
Plot korelasi pada Gambar 16 menunjukkan titik-titik atau dot-dot yang sangat rapat dan membentuk garis lurus, berbeda dengan plot korelasi

terhadap data uji karena model H2O AutoML dapat memprediksi dengan tepat dan nilai yang diprediksi tidak jauh berbeda dengan nilai asli pada data latih. Hal ini disebabkan oleh model memprediksi nilai atau value yang menjadi input untuk melakukan proses pelatihan ("*training*") model.



Gambar 17: Plot Garis Data Prediksi Terhadap Data Uji

Plot garis pada Gambar 17, hasil prediksi dari model H2O AutoML ditunjukkan oleh warna merah muda dan nilai asli dari data uji ditunjukkan oleh warna hijau. Terlihat bahwa hasil prediksi mengikuti pola yang ada pada data uji namun tidak persis sama dengan data uji, hal ini sudah dibuktikan pada evaluasi model yang menghasilkan tingkat akurasi model terhadap data uji yang mendapatkan nilai (R^2) 0.968 dan nilai *error* 3.48%.



Gambar 18: Plot Garis Data Prediksi Terhadap Data Latih

Plot garis pada Gambar 18 menunjukkan bahwa model H2O AutoML yang telah dibuat memiliki akurasi yang sangat tinggi dan *error* yang kecil sehingga data prediksi hampir sama persis terhadap data latih. Hal ini sudah dibuktikan pada evaluasi model yang menghasilkan tingkat akurasi model terhadap data latih yang mendapatkan nilai (R^2) 0.999 dan nilai error 1.38% yang disebabkan karena model memprediksi nilai atau value yang menjadi input untuk melakukan proses pelatihan ("*training*") model sehingga model dapat memprediksi dengan mudah nilai data latih. Namun plot garis ini juga menunjukkan bahwa model digunakan *overfitting* terhadap data yang digunakan. *Overfitting* adalah suatu keadaan dimana data yang digunakan untuk pelatihan itu adalah yang "terbaik". Sehingga apabila dilakukan tes dengan menggunakan data yang berbeda dapat mengurangi akurasi. Terjadinya *overfitting* saat memprediksi nilai pada data latih adalah hal wajar dan sering digunakan untuk membandingkan dengan model yang memprediksi terhadap data uji, *overfitting* yang tidak boleh dibiarkan adalah saat model memprediksi data uji karena model tersebut akan memiliki tingkat akurasi dan *error* yang sangat tinggi apabila menggunakan data yang berbeda atau belum ada.

Penutup

Model H2O AutoML telah berhasil dijalankan dengan model_id: StackedEnsemble_BestOfFamily_AutoML _20200617_063034 merupakan model yang baik. Berdasarkan hasil dari nilai statistik pada hasil memprediksi harga pembukaan Bitcoin model tersebut memperoleh nilai koefisien determinasi (R^2) sebesar 0.968 dan nilai *error* sebesar 3.48% terhadap data uji dan nilai koefisien determinasi (R^2) 0.999 dan nilai *error* sebesar 1.38% terhadap data latih, dapat dilihat bahwa model H2O AutoML memberikan tingkat akurasi yang cukup baik karena nilai R^2 cenderung mendekati 1 dan *error* hampir 0%.

Saran untuk pengembangan lebih lanjut untuk menyempurnakan dan memperluas penggunaan aplikasi yaitu dengan mencoba menggunakan dataset dari *cryptocurrency* lainnya, memperbarui dataset dan pengotomatisan aplikasi supaya dapat memprediksi harga Bitcoin atau uang digital lainnya setiap hari secara terus menerus.

Daftar Pustaka

[1] Rizky Amanda, Hasbi Yasin dan Alan Prahutama, "Analisis Support Vector Regression (SVR) dalam Memprediksi Kurs Rupiah terhadap Dollar Amerika Serikat", Jurnal Gaussian, vol. 3.,nomor 4, pp 849 sd 858, 2014.

[2] Novia Agustina, Suparti dan Moch Abdul Mukid, "Pemodelan Data Indeks Harga Saham Gabungan Menggunakan Regresi Penalized Spline", Jurnal Gaussian, vol. 4, nomor 3, pp 603 sd 612, 2015.

[3] Bagaskara Ramadhan, " Visualisasi Data Untuk Memprediksi Pasar Saham Dari Hasil Pengolahan Data Set S&P 500 Dengan Menggunakan R-Programming", Penulisan Ilmiah, Universitas Gunadarma, 2018.

[4] Felita Setiawan, "Coinvestasi: Apa itu Cyrptocurrency?", diakses daring pada https://coinvestasi.com /belajar/apa-itu-cryptocurrency, Online, diakses 10 April 2020.

[5] Anonim, "Situs resmi Bitcoin. Pertanyaan yang Sering Diajukan", diakses daring pada https://bitcoin.org/id/faq , Online, diakses 10 April 2020.

[6] Rian Adam, "Mengenal Google Colab", diakses daring pada https://structilmy.com/2019/05/mengenal-google-colab/ , Online, diakses 14 Mei 2020.

[7] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, "Bitcoin dan teknologi mata uang kripto: pengantar yang komprehensif", Princeton University Press. 2016.

[8] Naomi Ceder, "The Quick Phyton Book", Third Edition, Manning Publications, ISBN-13:978-1617294037, 2018.

[9] Anonym, "AutoML: Automatic Machine Learning", diakses daring pada http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html , Online, diakses 15 Mei 2020.

[10] AM. Ghofur, "Koefisisen Determinasi dalam Analisis Regresi", diakses daring pada https://maglearning.id /2019/02/24/koefisien-determinasi-dalam-analisis-regresi/ , Online, diakses 18 Juni 2020.

[11] Mohammad Shahebaz,"Gentle Introduction to AutoML from H2O.ai.", diakses daring pada .https://medium.com/analytics-vidhya/gentle-introduction-to-automl- from-h2o-ai-a42b393b4ba2 , Online, diakses 11 April 2020.

[12] Supardi, "Aplikasi Statistika dalam Penelitian Konsep Statistika yang Lebih Komprehensif", Jakarta: Change Publication, 2013.

[13] Jeny Purwati, "Likuiditas dan Efisiensi Pasar pada Mata Uang Kripto", Skripsi. Fakultas Ekonomi, Manajemen, Universitas Islam Indonesia, Yogyakarta. 2019.

[14] Hadari Nawawi, "Analisis Regresi dengan MS Excel 2007 dan SPSS 17", PT. Elex Media Komputindo, Jakarta, 2010.