

Prediksi Permintaan Kargo pada Cargo Service Center Tangerang City Menggunakan Metode Gated Recurrent Unit

Rifky Ahmad Saputra^{*1}, Yufis Azhar², Vinna Rahmayanti³

^{1,2,3}Teknik Informatika/Universitas Muhammadiyah Malang

rifkyahmadsaputra@gmail.com^{*1}, yufis.az@gmail.com², vinna.nastiti@gmail.com³

Abstrak

Pada saat ini persaingan bisnis dalam bidang layanan kargo khususnya di Indonesia semakin ketat. Terdapat beberapa perusahaan layanan kargo di Indonesia, salah satunya yaitu Cargo Service Center Tangerang City. Untuk mengantisipasi persaingan bisnis tersebut, Cargo Service Center Tangerang City harus dapat menentukan strategi manajemen usaha, baik dalam jangka menengah maupun jangka panjang. Salah satunya hal yang dapat dilakukan yaitu prediksi permintaan kargo. Pada Cargo Service Center Tangerang City terdapat data transaksi kargo mulai dari Januari 2016 hingga September 2019, oleh karena itu dilakukanlah penelitian yaitu mengimplementasikan metode Gated Recurrent Unit untuk melakukan prediksi permintaan kargo. metode Gated Recurrent Unit merupakan model pengembangan dari Recurrent Neural Network yang biasa digunakan untuk melakukan prediksi pada data sekuens. Pengujian model prediksi dalam penelitian ini dilakukan dengan mencari nilai Root Mean Square Error terkecil dari beberapa percobaan. Hasil dari penelitian ini menunjukkan bahwa model cukup baik dalam melakukan prediksi permintaan kargo, namun terdapat beberapa hasil prediksi metode Gated Recurrent Unit yang masih belum maksimal mendekati nilai aktual misalnya pada nilai aktual yang berada di titik puncak.

Kata Kunci: Kargo, Prediksi, Deep Learning, Recurrent Neural Network, Gated Recurrent Unit

Abstract

At this time, business competition in the field of cargo shipping especially in Indonesia is getting tougher. There are some cargo service in Indonesia, one of them is Cargo Service Center Tangerang City. To anticipate that business competition, Cargo Service Center Tangerang City must be able to determine business management strategies in the medium and long term. One of the things that can be done is predicting cargo demand. At Cargo Service Center Tangerang City, there are cargo transaction data from January 2016 to September 2019, therefore conduct research study that is implementing Gated Recurrent Unit method for predict cargo demand at Cargo Service Center Tangerang City. Gated Recurrent Unit is a development model of the Recurrent Neural Network that is commonly used to make predictions on sequence data. Testing prediction model in this research is conducted by finding the smallest Root Mean Square Error value from several experiments. The results of this research indicate that the model is quite good in predicting cargo demand, but there are some Gated Recurrent Unit prediction results that are still not maximally close to the actual value, for example at the actual value in the peak point.

Keywords: Cargo, Prediction, Deep Learning, Recurrent Neural Network, Gated Recurrent Unit

1. Pendahuluan

Permintaan kargo pada saat ini mengalami perkembangan yang cukup pesat di Indonesia, hal ini dikarenakan dengan pertumbuhan sosial ekonomi serta permintaan konsumen yang selalu meningkat. Kargo dapat didefinisikan sebagai semua barang yang dikirim melalui udara, darat maupun laut yang dilengkapi oleh dokumen pengangkutan (Air Way Bill) antar wilayah atau kota di dalam negeri maupun antarnegara [1]. Terdapat beberapa perusahaan layanan kargo di Indonesia, salah satunya yaitu CSC Tangerang City. CSC Tangerang City merupakan layanan kargo udara resmi dari Garuda yang berada pada Tangerang di Jalan Palem Raja Raya Selatan, Karawaci, Tangerang. CSC ini bertujuan sebagai tempat drop dan pick-up point barang, sehingga memudahkan pelanggan yang ingin mengirimkan barangnya tanpa perlu lagi harus datang ke bandara.

Pada saat ini persaingan bisnis dalam bidang layanan kargo khususnya di Indonesia semakin ketat, salah satunya diakibatkan oleh perusahaan layanan kargo yang bertambah banyak dikarenakan pertumbuhan sosial ekonomi dan permintaan konsumen untuk kargo mengalami peningkatan. Dalam jangka panjang perusahaan khususnya CSC Tangerang City harus dapat meningkatkan dan memantapkan posisi usahanya di tengah-tengah persaingan bisnis yang semakin ketat dan kompleks. Untuk mengantisipasi hal itu, CSC Tangerang City harus dapat menentukan strategi manajemen usaha, baik dalam jangka menengah maupun jangka panjang. Salah satu yang dapat dilakukan adalah prediksi permintaan kargo pada perusahaan tersebut menggunakan data-data transaksi yang telah direkap setiap harinya.

Pada CSC Tangerang City terdapat data transaksi kargo mulai dari Januari 2016 hingga September 2019, oleh karena itu dilakukanlah penelitian untuk prediksi permintaan kargo pada CSC Tangerang City dari data-data yang telah tersedia. Prediksi dilakukan dengan mengimplementasikan model Deep Learning, yaitu menggunakan metode Gated Recurrent Unit (GRU). Metode GRU-NN merupakan metode perkembangan dari metode Recurrent Neural Network yang dapat melakukan prediksi pada data time-series (deret waktu).

Pada penelitian Baxter, dkk [2] melakukan prediksi permintaan kargo udara ekspor Australia tahun dengan menggunakan metode Artificial Neural Network dan menghasilkan nilai Root Mean Square Error (RMSE) sebesar 1.3×10^4 . Pada penelitian Uyar, dkk [3] melakukan prediksi tarif permintaan angkutan kargo dengan menggunakan metode Fuzzy Recurrent Neural Network dengan menghasilkan nilai MAPE 14,96%. Sedangkan pada penelitian Fu, dkk [4] yaitu menggunakan metode ARIMA, LSTM dan GRU untuk prediksi arus lalu lintas. Hasil yang didapat adalah bahwa metode GRU lebih unggul nilai akurasi dibandingkan dari metode lainnya. Pada penelitian yang dilakukan Aji [5] melakukan prediksi pada data time series (deret waktu) yaitu melakukan prediksi harga bitcoin dengan menggunakan metode GRU. Penelitian tersebut menghasilkan nilai MAPE sebesar 5,43%.

Penelitian ini mengusulkan penggunaan metode Gated Recurrent Unit (GRU) untuk melakukan prediksi permintaan kargo. Metode GRU diperkenalkan pada tahun 2014 oleh Cho, dkk [6]. GRU merupakan metode pengembangan dari Recurrent Neural Network (RNN) yang bertujuan untuk membuat setiap recurrent unit bisa dapat menangkap hubungan (dependensi) dalam skala waktu yang berbeda-beda secara adaptif [7]. Menurut Rahman, dkk [8] menyatakan bahwa metode GRU memiliki lebih sedikit parameter daripada Long Short-Term Memory (LSTM) sehingga komputasinya lebih sederhana dan waktu yang dibutuhkan lebih sedikit. Serta akurasi yang didapat setara bahkan bisa lebih dari metode LSTM, tergantung dari data dan studi kasusnya. Berdasarkan hasil yang didapat pada penelitian Fu, dkk [4] menunjukkan bahwa metode GRU memiliki nilai kesalahan (error) yang lebih kecil dibandingkan metode LSTM dan ARIMA.

Berdasarkan dari penjelasan sebelumnya, maka dilakukan penelitian tentang prediksi permintaan kargo pada perusahaan CSC Tangerang City dengan menggunakan metode Gated Recurrent Unit (GRU). Data yang digunakan untuk penelitian ini adalah data transaksi kargo CSC Tangerang City dari Januari 2016 hingga September 2019. Pada penelitian ini diharapkan dapat mengimplementasikan metode GRU dalam melakukan prediksi permintaan kargo pada CSC Tangerang City dan menghasilkan nilai kesalahan (error) yang kecil.

2. Metode Penelitian

2.1 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data mentah transaksi kargo pada perusahaan Cargo Service Center Tangerang City mulai dari Januari 2016 hingga September 2019. Data mentah tersebut berisikan kumpulan transaksi pengiriman kargo dalam perhari yang direkap tiap bulan.

2.2 Preprocessing Data

Preprocessing data adalah sebuah proses yang dilakukan untuk mempersiapkan data mentah yang telah dikumpulkan agar menjadi data yang berkualitas. Pada *preprocessing* data terdapat beberapa tahap yang dilakukan, tahap tersebut dijelaskan sebagai berikut.

2.2.1 Merubah Format Data

Data yang telah dikumpulkan akan dirubah format datanya dari format data pertransaksi menjadi perhari dengan cara melakukan subtotal pada transaksi (data numerik) berdasarkan

tanggalnya. Setelah itu, melakukan penyamaan format tanggal karena pada data mentah tersebut berisikan format tanggal yang berbeda-beda (tidak konsisten). Setelah data telah diubah formatnya, maka dipilih beberapa fitur yang akan digunakan dalam melakukan prediksi permintaan kargo.

2.2.2 Mengatasi Data *Outlier*

Pada tahap ini, dalam menentukan suatu data merupakan data noise (*outlier*) atau tidak dapat dilakukan dengan menggunakan Boxplot. Boxplot merupakan ringkasan distribusi sampel yang disajikan secara grafis yang dapat menggambarkan distribusi data, ukuran tendensi sentral dan ukuran penyebaran. Boxplot juga dapat menunjukkan ada tidaknya nilai *outlier* dari data tersebut.

2.2.3 Transformasi Logaritma Natural

Setelah data bersih dari *outlier*, dilakukan transformasi logaritma natural untuk menstabilkan dan memperkecil nilai variansi dari data tersebut. Dalam melakukan transformasi logaritma natural pada data, dapat dilakukan dengan Persamaan 1.

$$y' = \ln(y) \quad (1)$$

2.2.4 Normalisasi *Min-Max*

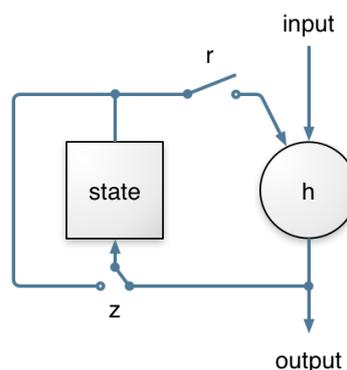
Setelah melakukan transformasi, dilanjutkan dengan melakukan normalisasi data dengan menggunakan metode normalisasi min-max. Metode min-max adalah metode normalisasi dengan melakukan transformasi linier terhadap data asli, seperti pada Persamaan 2 [9].

$$x' = \frac{x - \min(x)}{\max(X) - \min(x)} \quad (2)$$

2.3 Prediksi Menggunakan Metode Gated Recurrent Unit

Pada tahap ini melakukan prediksi dari dataset transaksi yang telah di preprocessing dengan menggunakan metode Gated Recurrent Unit (GRU). GRU merupakan metode pengembangan dari Recurrent Neural Network (RNN) yang bertujuan untuk membuat setiap recurrent unit bisa dapat menangkap hubungan (dependensi) dalam skala waktu yang berbeda-beda secara adaptif. Secara analogi, metode GRU menggunakan beberapa informasi pada masa lalu (sebelumnya) yang relevan untuk dapat membuat keputusan sekarang [6]. Ilustrasi dari metode GRU dapat dilihat pada Gambar 1.

Gated recurrent unit (GRU)



Gambar 1. Ilustrasi dari Metode GRU

Pada Gambar 1, merupakan ilustrasi dari metode GRU. Pada Metode GRU terdapat dua gerbang yaitu reset gate 'r' dan update gate 'z' yang digunakan untuk membentuk hidden state 'h'. Update gate digunakan untuk menentukan berapa banyak informasi masa lalu (sebelumnya) yang harus tetap disimpan. Sedangkan reset gate digunakan untuk memutuskan berapa banyak informasi masa lalu untuk dihapus dan menentukan bagaimana menggabungkan input baru

dengan informasi masa lalu (sebelumnya) [7]. Persamaan 3, Persamaan 4, dan Persamaan 5 berikut merupakan rumus-rumus yang terdapat pada metode GRU.

1. *Update Gate*

$$z_t = \sigma_g (W_z x_t + U_z h_{t-1} + b_z) \quad 3$$

2. *Reset Gate*

$$r_t = \sigma_g (W_r x_t + U_r h_{t-1} + b_r) \quad 4$$

3. *Hidden State*

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h (W_h x_t + U_h (r_t \circ h_{t-1}) + b_h) \quad 5$$

2.4 Pengujian dan Evaluasi Model Prediksi

Setiap model prediksi yang telah dibuat, harus dilakukan pengujian dan evaluasi hasil prediksinya untuk mengetahui performa dari model prediksi tersebut. Dalam pengujian model prediksi, melakukan beberapa percobaan dengan nilai parameter yang berbeda-beda untuk menghasilkan hyperparameter terbaik untuk model prediksi. Dalam mengukur performa model prediksi, digunakan metode perhitungan Root Mean Square Error (RMSE) yang merupakan nilai akar dari nilai rata-rata error yang telah dikuadratkan. Nilai RMSE rendah menunjukkan bahwa variasi nilai hasil prediksi (output) mendekati variasi nilai sebenarnya (target) [10]. Persamaan 6 berikut merupakan rumus dari RMSE.

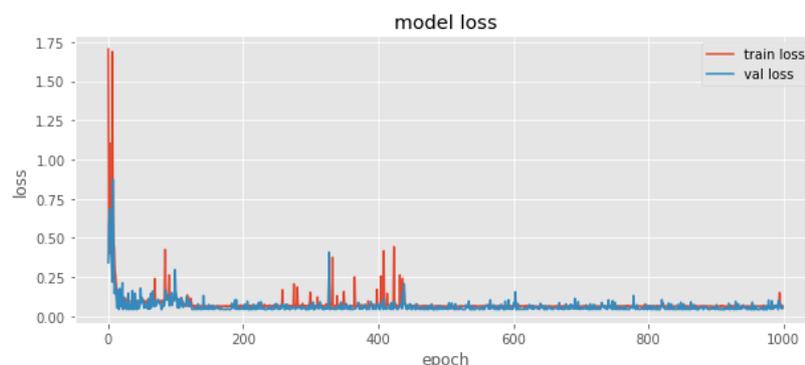
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad 6$$

3. Hasil Penelitian dan Pembahasan

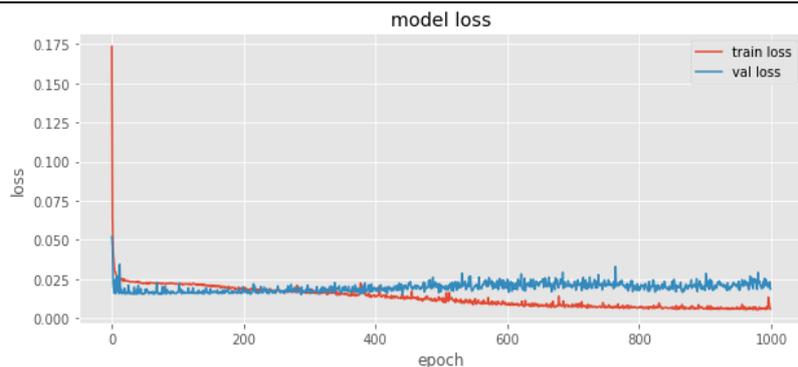
Implementasi dilakukan untuk memastikan sistem dapat berjalan sesuai dengan yang diharapkan. Implementasi dimulai dengan melakukan preprocessing data agar dapat digunakan, setelah itu mengimplementasikan arsitektur metode Gated Recurrent Unit untuk membuat model prediksi dan diakhiri dengan melakukan pengujian untuk mencari tahu hyperparameter terbaik berdasarkan hasil evaluasi dari nilai Root Means Square Error (RMSE) model tersebut.

3.1 Pengujian Penentuan *Learning Rate*

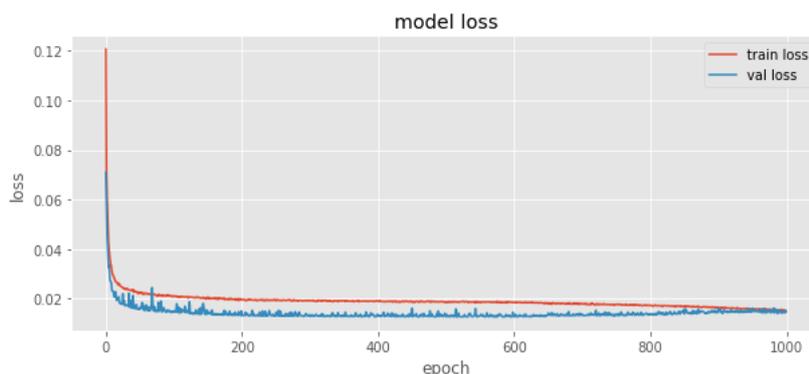
Learning Rate adalah nilai koreksi pembaharuan bobot dalam model prediksi, learning rate merupakan salah satu hyperparameter yang diujikan dalam penelitian ini. Pada pengujian ini, penentuan nilai *learning rate* terbaik ditentukan berdasarkan grafik model *loss* yang tidak mengalami fluktuasi signifikan, tidak terjadi *overfitting* pada data validasi dan menghasilkan nilai validasi *loss* yang paling kecil. Pengujian *learning rate* yang akan diujikan adalah 0.1, 0.01 dan 0.001. Berikut merupakan grafik model *loss* hasil pengujian yang dilakukan, dapat dilihat pada Gambar 2, Gambar 3, dan Gambar 4.



Gambar 2. Grafik Model Loss dengan Nilai Learning Rate 0.1



Gambar 3. Grafik Model Loss dengan Nilai Learning Rate 0.01



Gambar 4. Grafik Model Loss dengan Nilai Learning Rate 0.001

Pada Gambar 2 dan Gambar 3, model prediksi dengan nilai *learning rate* 0.1 dan 0.01 terjadi fluktuasi terhadap *training loss* dan *validation loss*. Hal tersebut dapat dinyatakan bahwa nilai *learning rate* terlalu besar bagi model prediksi. Sedangkan pada Gambar 4 dengan nilai *learning rate* tidak mengalami fluktuasi pada *training loss* maupun model *loss* serta tidak mengalami *overfitting* pada model prediksi. Berdasarkan beberapa pengujian yang dilakukan, berikut merupakan hasil *training loss* dan *validation loss* dari model prediksi pada masing-masing nilai *learning rate* yang dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Penentuan Learning Rate

Learning rate	Training loss	Validation loss
0.1	0.06619	0.05541
0.01	0.005956	0.018616
0.001	0.015063	0.014608

Berdasarkan pada Tabel 1, dapat dilihat bahwa nilai *training loss* dan *validation loss* pada pengujian *learning rate* 0.1 paling besar. Sedangkan pada pengujian nilai *learning rate* 0.01 menghasilkan nilai *training loss* yang sangat kecil, namun di sisi lain nilai *validation loss* nya masih besar hal ini menandakan bahwa terjadinya *overfitting* pada model. Pada pengujian nilai *learning rate* 0.001 menghasilkan nilai *training loss* dan *validation loss* yang sama-sama kecil, hal tersebut dapat dinyatakan nilai *learning rate* tersebut cocok digunakan pada model prediksi ini. Berdasarkan hasil dari pengujian yang dilakukan, maka nilai *learning rate* yang digunakan pada model prediksi ini adalah 0.001.

3.2 Pengujian Penentuan Hidden Unit

Pengujian ini dilakukan untuk mendapatkan jumlah *hidden unit* terbaik pada model prediksi. Jumlah *hidden unit* yang diujikan pada penelitian ini yaitu 32, 64, 128 dan 256. Penentuan jumlah *hidden unit* terbaik didasarkan pada nilai rata-rata RMSE terkecil dengan masing-masing jumlah *hidden unit* diuji menggunakan nilai *learning rate* 0.001 dan jumlah *batch*

size yang berbeda-beda yaitu 2, 4, 8 dan 16. Berikut merupakan hasil dari pengujian penentuan jumlah *hidden unit* yang dapat dilihat pada Tabel 2, Tabel 3, Tabel 4, dan Tabel 5.

Tabel 2. Hasil RMSE dengan Jumlah Hidden Unit 32

<i>Hidden unit</i>	<i>Learning Rate</i>	<i>Batch size</i>	RMSE
32	0.001	2	313.261
32	0.001	4	298.1643
32	0.001	8	326.9122
32	0.001	16	247.3952
Rata-rata RMSE :			296.4333

Tabel 3. Hasil RMSE dengan Jumlah Hidden Unit 64

<i>Hidden unit</i>	<i>Learning Rate</i>	<i>Batch size</i>	RMSE
64	0.001	2	353.5218
64	0.001	4	298.7832
64	0.001	8	279.5317
64	0.001	16	274.184
Rata-rata RMSE			301.5

Tabel 4. Hasil RMSE dengan Jumlah Hidden Unit 128

<i>Hidden unit</i>	<i>Learning Rate</i>	<i>Batch size</i>	RMSE
128	0.001	2	280.7765
128	0.001	4	267.1991
128	0.001	8	313.0491
128	0.001	16	353.0937
Rata-rata RMSE			303.5295

Tabel 5. Hasil RMSE dengan Jumlah Hidden Unit 256

<i>Hidden unit</i>	<i>Learning Rate</i>	<i>Batch size</i>	RMSE
256	0.001	2	337.6409
256	0.001	4	279.5032
256	0.001	8	288.1624
256	0.001	16	304.8415
Rata-rata RMSE			302.537

Berdasarkan hasil pengujian pada Tabel 2, Tabel 3, Tabel 4, dan Tabel 5, dapat dilihat nilai rata-rata RMSE terkecil terdapat pada jumlah *hidden unit* sebesar 32 yaitu 296.4333. Sedangkan nilai RMSE terkecil terdapat pada jumlah *learning rate* 0.001, *hidden unit* sebesar 32 dan *batch size* sebesar 16 dengan menghasilkan RMSE sebesar 247.3952. Berdasarkan hasil rata-rata RMSE pada beberapa pengujian tersebut, dapat dinyatakan bahwa jumlah *hidden unit* yang digunakan pada penelitian ini yaitu sebesar 32.

3.3 Pengujian Penentuan *Batch Size*

Pada tahap ini, melakukan pengujian untuk mendapatkan jumlah *batch size* terbaik pada model prediksi. Jumlah *batch size* yang diujikan pada penelitian ini yaitu 2, 4, 8 dan 16. Penentuan pemilihan jumlah *batch size* terbaik didasarkan pada nilai rata-rata RMSE terkecil yang diuji menggunakan variasi jumlah *epoch* yang berbeda-beda yaitu 1000, 1500 dan 2000. Berikut merupakan hasil dari pengujian penentuan *hidden unit* yang dapat dilihat pada Tabel 6, Tabel 7, Tabel 8, dan Tabel 9.

Tabel 6. Hasil RMSE dengan Jumlah Batch Size 2

Batch size	Learning Rate	Hidden Unit	Epoch	RMSE
2	0.001	16	1000	313.261
2	0.001	16	1500	287.4729
2	0.001	16	2000	294.3051
Rata-rata RMSE				223.7598

Tabel 7. Hasil RMSE dengan Jumlah Batch Size 4

Batch size	Learning Rate	Hidden Unit	Epoch	RMSE
4	0.001	16	1000	298.1643
4	0.001	16	1500	309.8785
4	0.001	16	2000	296.8327
Rata-rata RMSE				226.2189

Tabel 8. Hasil RMSE dengan Jumlah Batch Size 8

Batch size	Learning Rate	Hidden Unit	Epoch	RMSE
8	0.001	16	1000	326.9122
8	0.001	16	1500	309.3132
8	0.001	16	2000	344.5636
Rata-rata RMSE				245.1973

Tabel 9. Hasil RMSE dengan Jumlah Batch Size 16

Batch size	Learning Rate	Hidden Unit	Epoch	RMSE
16	0.001	16	1000	247.3952
16	0.001	16	1500	268.9501
16	0.001	16	2000	297.2386
Rata-rata RMSE				203.396

Berdasarkan hasil pengujian pada Tabel 6, Tabel 7, Tabel 8, dan Tabel 9, dapat dilihat nilai rata-rata RMSE terkecil terdapat pada jumlah *batch size* sebesar 16 yaitu 203.396. Sedangkan nilai RMSE terkecil terdapat pada komposisi *hyperparameter learning rate* sebesar 0.001, *hidden unit* sejumlah 32, *batch* sejumlah 16 dan *epoch* sejumlah 1000 dengan menghasilkan RMSE sebesar 247.3952. Berdasarkan hasil rata-rata RMSE pada beberapa pengujian tersebut, dapat dinyatakan bahwa jumlah *batch size* yang digunakan pada penelitian ini yaitu sebesar 16.

3.4 Pengujian Penentuan *Epoch* dan Partisi Data

Pengujian pada tahap ini dilakukan untuk mengetahui jumlah *epoch* dan partisi data yang terbaik pada model prediksi. Jumlah *epoch* yang diujikan pada penelitian ini yaitu 2, 4, 8 dan 16 sedangkan partisi data yang diujikan adalah data latih 60%, data validasi 10% dan data uji 30%, data latih 70%, data validasi 10% dan data uji 20% dan data latih 80%, data validasi 10% dan data uji 10%. Penentuan pemilihan jumlah *epoch* dan partisi data terbaik didasarkan pada nilai RMSE terkecil. Berikut merupakan hasil dari pengujian penentuan *hidden unit* yang dapat dilihat pada Tabel 10, Tabel 11, dan Tabel 12.

Tabel 10. Hasil RMSE dengan Epoch 1000 dan Partisi Data Bervariasi

Epoch	Partisi Data	Learning Rate	Hidden Unit	Batch size	RMSE
1000	Data latih=60% Data validasi=10% Data uji=30%	0.001	32	16	307.419
1000	Data latih=70% Data validasi=10% Data uji=20%	0.001	32	16	247.3952
1000	Data latih=80% Data validasi=10% Data uji=10%	0.001	32	16	260.659

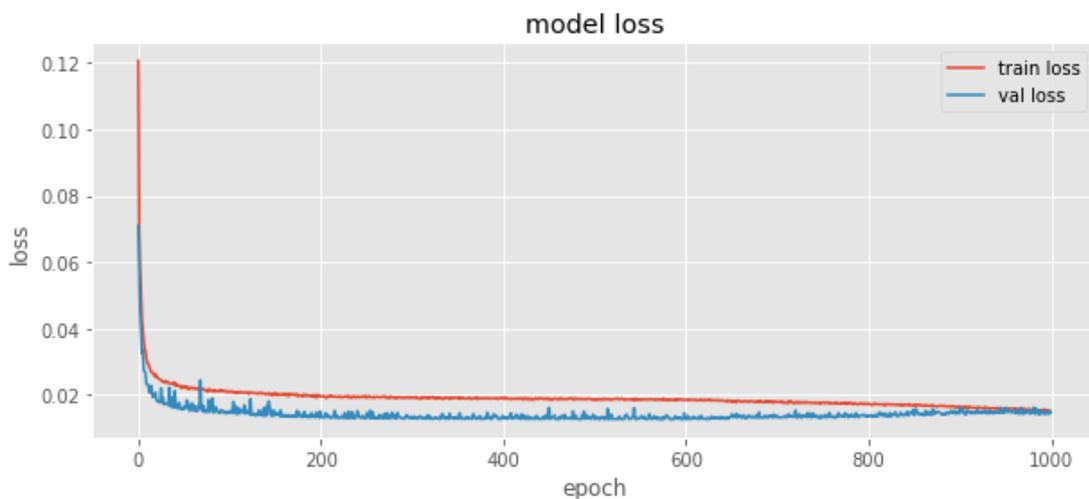
Tabel 11. Hasil RMSE dengan Epoch 1500 dan Partisi Data Bervariasi

Epoch	Partisi Data	Learning Rate	Hidden Unit	Batch size	RMSE
1500	Data latih=60% Data validasi=10% Data uji=30%	0.001	32	16	314.4545
1500	Data latih=70% Data validasi=10% Data uji=20%	0.001	32	16	268.9501
1500	Data latih=80% Data validasi=10% Data uji=10%	0.001	32	16	324.8828

Tabel 12. Hasil RMSE dengan Epoch 2000 dan Partisi Data Bervariasi

Epoch	Partisi Data	Learning Rate	Hidden Unit	Batch size	RMSE
2000	Data latih=60% Data validasi=10% Data uji=30%	0.001	32	16	352.6851
2000	Data latih=70% Data validasi=10% Data uji=20%	0.001	32	16	297.2386
2000	Data latih=80% Data validasi=10% Data uji=10%	0.001	32	16	358.3605

Berdasarkan hasil pengujian pada Tabel 10, Tabel 11, dan Tabel 12, dapat ditetapkan *hyperparameter* terbaik dari model prediksi pada penelitian ini ialah dengan komposisi sebagai berikut *learning rate* sebesar 0.001, *hidden unit* sejumlah 32, *batch* sejumlah 16, *epoch* sejumlah 1000 dan partisi data dengan komposisi data latih sebesar 70%, data validasi sebesar 10% dan data uji sebesar 20%. Hasil RMSE yang dihasilkan oleh model prediksi dengan *hyperparameter* tersebut adalah sebesar 247.3952. Berikut merupakan grafik model *loss* yang dihasilkan oleh model prediksi pada penelitian ini yang dapat dilihat pada Gambar 5.

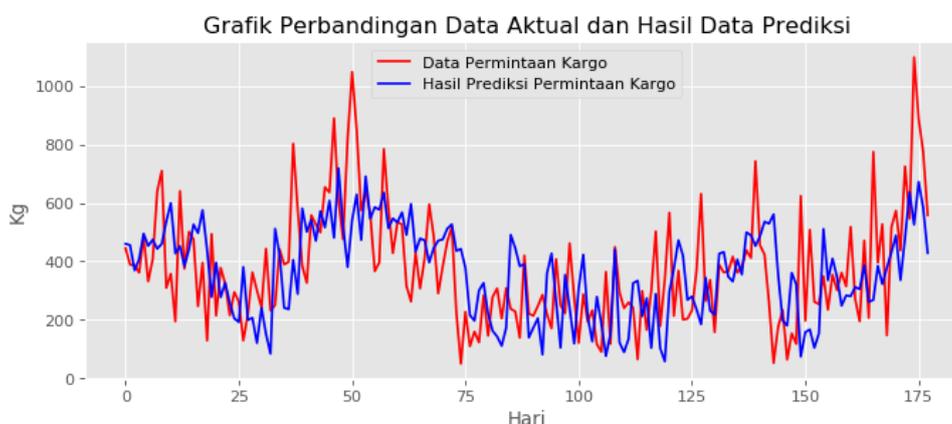


Gambar 5. Grafik Model Loss dari Model Prediksi Permintaan Kargo

Berdasarkan Gambar 5, grafik yang dihasilkan sangat baik tidak terdapat fluktuasi yang signifikan serta tidak mengalami *overfitting* pada model prediksi. Berikut merupakan sampel hasil dan grafik perbandingan data aktual dan data prediksi yang dapat dilihat pada Tabel 13 dan Gambar 6.

Tabel 13. Sampel Hasil Perbandingan Data Aktual dan Data Prediksi

Data Aktual	Data Prediksi
444.5	459.865814
388.5	455.405029
387.0	369.029907
361.0	407.359467
484.5	494.883942
...	...
546.0	637.285767
1099.0	525.613159
889.4	672.127136
776.0	583.760986
558.0	428.463043



Gambar 6. Grafik Perbandingan Data Aktual dan Hasil Data Prediksi

Dapat dilihat pada Tabel 13 dan Gambar 6, selisih data aktual dan data prediksi tidak terlalu besar. Dapat dilihat grafik tersebut bahwa pergerakan data hasil prediksi dapat mengikuti pola pergerakan data aktual. Namun terdapat di beberapa titik misalnya dititik puncak, hasil prediksi GRU masih belum maksimal mengikuti pergerakan data aktual tersebut

4. Kesimpulan

Berdasarkan perancangan, implementasi dan hasil dalam penelitian ini, penentuan *hyperparameter* terbaik pada model prediksi dilakukan dengan melakukan beberapa pengujian berdasarkan nilai *hyperparameter* yang ditentukan. *hyperparameter* terbaik dari model prediksi pada penelitian ini adalah dengan komposisi : *learning rate* sebesar 0.001, *hidden unit* sejumlah 32, *batch* sejumlah 16, *epoch* sejumlah 1000 dan partisi data dengan komposisi data latih 70%, data validasi 10% dan data uji 20%. Hasil nilai kesalahan (*error*) RMSE dari model prediksi dengan menggunakan *hyperparameter* terbaik adalah sebesar 247.3952.

Berdasarkan hasil tersebut, model prediksi GRU cukup baik dalam melakukan prediksi permintaan kargo CSC Tangerang City, namun terdapat beberapa hasil prediksi metode GRU yang masih belum maksimal mendekati nilai aktual misalnya pada nilai aktual yang berada di titik puncak. Hal tersebut dikarenakan data yang digunakan masih sedikit serta nilai variansi data yang besar, sehingga model GRU belum maksimal dalam melakukan prediksi. Adapun beberapa hal yang dapat dilakukan dalam mengembangkan dan melakukan penelitian selanjutnya. Berikut merupakan beberapa saran dari penulis untuk penelitian selanjutnya :

1. Dapat menggunakan metode *Deep Learning* lainnya seperti LSTM, CNN dan lain-lain.
2. Menambahkan pengujian *hyperparameter* yang lainnya, seperti penentuan nilai lag, algoritma optimisasi (*optimizers*) yang digunakan dan lain-lain.
3. Menggunakan jumlah data yang lebih banyak, dikarenakan jumlah data berpengaruh dalam prediksi menggunakan metode GRU atau *Deep Learning*.

Daftar Notasi

y'	: Data hasil transformasi logaritma natural
\ln	: Logaritma natural
x'	: Data hasil normalisasi <i>min-max</i>
$\min(x)$: Nilai data <i>minimum</i>
$\max(X)$: Nilai data <i>maximum</i>
z_t	: <i>Output</i> dari proses <i>update gate</i>
σ_g	: Fungsi sigmoid
x	: Data <i>input</i>
W & U	: Nilai bobot
h_{t-1}	: Nilai <i>output</i> dari perhitungan sebelumnya
b	: Nilai bias
r_t	: <i>Output</i> dari proses <i>reset gate</i>
h_t	: Nilai <i>output</i> dari <i>hidden state</i>
y	: Nilai sebenarnya
\hat{y}_t	: Nilai hasil prediksi
n	: Banyak data

Referensi

- [1] A. Gusfadilah, B. D. Setiawan, and B. Rahayudi, "Implementasi Metode Exponential Smoothing Untuk Prediksi Bobot Kargo," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 2, pp. 1875–1882, 2019.
- [2] G. Baxter and P. Srisaeng, "The Use of an Artificial Neural Network to Predict Australia'S Export Air Cargo Demand," *Int. J. Traffic Transp. Eng.*, vol. 8, no. 1, pp. 15–30, 2018.
- [3] K. Uyar, "Long Term Dry Cargo Freight Rates Forecasting by Using Recurrent Fuzzy Neural Networks," in *12th International Conference on Application of Fuzzy Systems and Soft Computing*, 2016, pp. 642–647.
- [4] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction," in *31st Youth Academic Annual Conference of Chinese Association of Automation*, 2016, pp. 5–9.
- [5] Aji, Hendridha Wisnu. 2019. Skripsi. *Prediksi Harga Bitcoin dengan Menggunakan Metode Gated Recurrent Unit (GRU)*. Fakultas Matematika dan Ilmu Pengetahuan Alam. Jurusan Ilmu Komputer. Universitas Lambung Mangkurat. Banjarbaru.
- [6] K. Cho, D. Bahdanau, and F. Bougares, "Learning Phrase Representations using RNN Encoder – Decoder for Statistical Machine Translation," in *Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.
- [7] J. Chung, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," in *NIPS 2014 Workshop on Deep Learning*, 2014, pp. 1–9.
- [8] M. O. Rahman, S. Hossain, T. Junaid, and S. A. Forhad, "Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 1, pp. 213–222, 2019.
- [9] Hasihi, Amalia Erman. 2019. Skripsi. *Forecast Permintaan Kargo Udara dari Makassar ke Wilayah Papua Menggunakan Metode Double Constraint Gravity Model (DCGR)*. Jurusan Manajemen. Sekolah Tinggi Ilmu Ekonomi Nobel Indonesia. Makassar.
- [10] Rizki, Muhammad. 2019. Skripsi. *Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory (LSTM) untuk Prediksi Curah Hujan Kota Malang*. Fakultas Teknik. Jurusan Teknik Informatika. Universitas Muhammadiyah Malang. Malang.