J. Ris. & Ap. Mat. Vol. 10 No. 1 (2026) pp. 97-110
**Jurnal Riset dan Aplikasi Matematika (JRAM)**
e-ISSN: 2581-0154
DOI: doi.org/10.26740/jram.v10n1.p97-110

# PASSENGER FORECAST DOMESTIC FLIGHTS USING METHOD GATED RECURRENT UNIT

ASIH PURWANINGSI*, RIRI SYAFITRI LUBIS, AND R. MAISAROH REZYEKIYAH SIREGAR

Mathematics Study Program, Faculty of Science and Technology, State Islamic University of North Sumatra, Medan, Indonesia

*Corresponding Author: asih0703211004@uinsu.ac.id

***ABSTRACT***

*The number of air passengers is a crucial indicator in air transportation planning and management. However, passenger data often exhibits complex characteristics such as seasonality and fluctuations, making accurate forecasting a challenging task. Conventional statistical methods have limitations in capturing nonlinear patterns, thus requiring more advanced approaches. This study aims to develop a forecasting model for domestic air passengers at Kualanamu International Airport using the Gated Recurrent Unit (GRU) method. GRU is selected due to its ability to efficiently model sequential data, overcome the vanishing gradient problem in traditional Recurrent Neural Networks (RNN), and provide a simpler architecture compared to Long Short-Term Memory (LSTM). The dataset consists of monthly passenger data from January 2019 to November 2024. A sliding window approach with a window size of 12 is applied, and the model uses a stacked GRU architecture optimized with Adam and Mean Squared Error (MSE). The results show that the model achieves a Mean Absolute Percentage Error (MAPE) of 10.10% and Root Mean Square Error (RMSE) of 30,121, indicating good forecasting performance. The novelty of this study lies in the implementation of a multi-layer GRU model combined with seasonal feature inputs. The model is further used to predict passenger numbers for the next 24 months, showing consistent seasonal patterns.*

***Keywords:*** *Deep Learning, Forecasting, Gated Recurrent Unit, (GRU), Kualanamu Airport*

## 1. Introduction

Indonesia is an archipelagic country with geographically dispersed regions, requiring an efficient transportation system to support population mobility. An effective transportation system should ensure comfort, safety, and speed. In this context, air transportation plays a vital role as it enables fast and efficient connectivity between regions. Furthermore, air transportation contributes significantly to national economic growth and must be continuously improved in terms of infrastructure and service quality to enhance public welfare [1].

As one of the main modes of transportation, the number of air passengers in Indonesia fluctuates over time. These fluctuations are influenced by seasonal factors such as Eid al-Fitr, Eid al-Adha, Christmas, and New Year holidays. Kualanamu International Airport, as the main gateway in North Sumatra, serves millions of passengers annually. However, the COVID-19 pandemic caused a significant decline in passenger numbers, dropping from 3,191,442 passengers in 2018 to 2,180,199 passengers in 2019. Recent data from the Centramonth butcs Agency (BPS) shows that domestic passengers reached 205,372 in September 2024, increased

2020 *Mathematics Subject Classification*: 62M10, 68T07, 90B06.
Received: 01-03-2026; revised: 17-04-2026; accepted: 24-04-2026

by 7.94% from the previous month, but decreased again by 13.68% in October 2024 to 177,282 passengers. These fluctuations create challenges in airport capacity planning and operational management, highlighting the need for accurate forecasting methods.

Forecasting is a technique used to estimate future values based on historical data. In quantitative approaches, forecasting methods are generally categorized into causal methods, time series methods, and regression-based methods [2]. Time series methods are widely used because they can capture historical patterns systematically. However, traditional statistical models such as ARIMA and exponential smoothing have limitations in handling complex and nonlinear patterns in real-world data. With the advancement of computational technology, deep learning approaches have been increasingly applied in time series analysis. One of the most widely used architectures is the Recurrent Neural Network (RNN), which can process sequential data by utilizing information from previous time steps. However, RNN has limitations in capturing long-term dependencies due to the vanishing gradient and exploding gradient problems [3].

To overcome these limitations, the Gated Recurrent Unit (GRU) was introduced. GRU incorporates gating mechanisms, namely the reset gate and update gate, which enable the model to retain relevant information and discard unnecessary information. Compared to Long Short-Term Memory (LSTM), GRU has a simpler structure with fewer parameters, making it more computationally efficient while maintaining comparable performance in modeling sequential data [4]. Several studies have demonstrated the effectiveness of GRU in forecasting tasks, such as cargo demand prediction [5], palm oil price prediction [6], and stock price prediction [7]. However, most previous studies focus on general datasets and often use simple model architecture without incorporating additional features such as seasonality.

Despite these advancements, research specifically applying GRU to forecast domestic air passenger numbers at Kualanamu International Airport remains limited. In addition, previous studies rarely explore the use of multi-layer GRU architectures combined with seasonal features to improve forecasting accuracy. Therefore, this study proposes a multi-layer GRU model combined with seasonal feature inputs to address these limitations. The novelty of this research lies in the development of a more robust forecasting model that is capable of capturing both nonlinear patterns and seasonal characteristics in passenger data. This approach is expected to provide more accurate predictions and support better decision-making in airport management.

## 2. Literature Review

### 2.1. Forecasting

Forecasting is the process of estimating or predicting future events or values based on historical data, patterns, and relevant information. According to Hyndman & Athanasopoulos (2018), forecasting is an important tool in decision-making, especially in business, economics, and operations management. Forecasting helps organizations plan resources, manage inventory, and anticipate market changes [8].

### 2.2. Passenger

A passenger is anyone who uses transportation services such as airplanes or other means of transportation based on an agreement or consent with the transportation service provider, whether a company or authorized institution.

### 2.3. Aircraft

Airplanes are a mode of air transportation used to transport passengers relatively quickly from one place to another [9]. Airplanes are a means of transportation used by people to support various activities, both in business and tourism.

### 2.4. Domestic Flights

Based on Law Number 1 of 2019 concerning aviation, Article 1 paragraph 1, aviation is a unified system consisting of the use of airspace, aircraft, airports, air transportation, flight navigation, safety and security, the environment, as well as supporting facilities and other public facilities.

### 2.5. *Gated Recurrent Unit* (GRU)

The advantages of the Gated Recurrent Unit (GRU) are more efficient memory usage and its effective ability to process sequential data [10]. According to Halim, et al., (2022), the initial steps in the Gated Recurrent Unit (GRU) method are:

1. *Reset Gate*

The calculation of the reset gate can be seen in equation 1:

$$r = \sigma(w_r x_t + u_r h_{t-1} + b_r) \quad (1)$$

Description:

$r$ : Reset gate
$\sigma$ : Sigmoid activation function
$x_t$ : Input data
$w_r$ : Weight parameter
$h_{t-1}$ : Hidden state from the previous time step
$u_r$ : Weight parameter
$r$ : Bias value at the reset gate

2. *Update Gate*

The calculation for the update gate can be seen in equation 2:

$$z = \sigma(w_z x_t + u_z h_{t-1} + b_z) \quad (2)$$

Description:

$z$ : Update gate
$\sigma$ : Sigmoid activation function
$x_t$ : Input data
$w_z$ : Weight parameter
$h_{t-1}$ : Hidden state from the previous time step
$b_r$ : Bias value at the update gate

3. Hidden State Candidate

The calculation of the hidden state candidate can be seen in equation 3:

$$\tilde{h} = \tanh(w_h x_t + u_h(r h_{t-1}) + b_h) \quad (3)$$

Description:

$\tilde{h}$ : Candidate hidden state
$r$ : Reset gate
$x_t$ : Input data
$w_h$ : Weight parameter
$h_{t-1}$ : Hidden state from the previous time step
$b_h$ : Bias value

4. *Hidden State*

The calculation of the hidden state can be seen in equation 4:

$$h = (1 - z)\tilde{h} + z h_{t-1} \quad (4)$$

Description:

$h$ : Output
$\tilde{h}$ : Candidate hidden state
$z$ : Output of the update gate

$h_{t-1}$ : Hidden state from the previous time step

## 2.6. *Sliding Window Algorithm* (SWA)

Windowing is the process of creating a structure from existing time series data. Before data is processed using the Gated Recurrent Unit (GRU) algorithm, it must undergo preprocessing. One of the preprocessing steps is window size. Preprocessing consists of two stages: data splitting and data normalization. In the data splitting stage, the data is divided into training and testing data. Next, the normalization stage occurs. The goal of data normalization is to reduce large data variations, as these variations can affect prediction results [11].

The Min-Max Normalization formula can be calculated as follows:

$$z = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{5}$$

Description:

$z$ : Normalized value
$x_i$ : i-th data
$x_{\min}$ : Smallest value
$x_{\max}$ : Largest value

## 2.7. *Hypertuning Parameter*

Hypertuning Parameter is one of the factors that influence model performance to get better results when training the model. The parameters used in the hypertuning process are learning gate, hidden state, adam optimization, batch size, window size and epoch [12]. Hypertuning Parameter is one of the factors that influence model performance to get better results when training the model. The parameters used in the hypertuning process are learning gate, hidden state, adam optimization, batch size, window size and epoch [12].

## 2.8. Activation Function

The activation functions that are frequently used are the sigmoid and tanh functions as follows:

1. The sigmoid activation function is a non-linear function that has a range of 0 to 1. The sigmoid function equation can be seen as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

Description:

$x$ : Input data
$e$ : Constant (approximately 2.7182)

2. The tanh activation function or hyprotbolic tangent is a value that changes in the range -1 to 1. The tanh function equation can be seen as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{7}$$

## 2.9. Denormalization

Denormalization aims to return data to its original value form, so that the resulting output becomes easier to read and understand. The calculation of the denormalization process can be seen in Equation [2]. Denormalization aims to return data to its original value form, so that the resulting output becomes easier to read and understand. The calculation of the denormalization process can be seen in Equation [2]:

$$d = d'(\max - \min) + \min \tag{8}$$

Description:

$d$ : Denormalization result value

$d'$ : Normalized data value
max : Maximum value of actual data
min : Minimum value of actual output data

### 2.10. Model Evaluation

To evaluate the performance of the model in this study, Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) were used [13].

1. Root Mean Square Error (RMSE)

   The Root Mean Square Error (RMSE) is a measure of the degree of error in a prediction. The smaller the RMSE, the more accurate the prediction is, as it approaches zero. The formula is as follows.

$$RMSE = \frac{\sqrt{\sum_i^n \left(\widehat{Y}_t - Y_t\right)^2}}{n} \tag{9}$$

Description:
$Y_t$ : Actual value for period t
$\widehat{Y}_t$ : Predicted value for period t
$n$ : Number of data

2. Mean Absolute Percentage Error (MAPE)

   Mean Absolute Percentage Error (MAPE) is the average of the absolute differences between the forecasted value and the actual value, expressed as a percentage of the actual value. The use of MAPE in evaluating prediction results is useful for seeing how accurate the forecast results are compared to the actual data [14].

$$MAPE = \frac{\sum_{t=1}^{n} \left|\frac{Y_t - \widehat{Y}_t}{Y_t}\right|}{n} \times 100\% \tag{10}$$

Description:
$Y_t$ : Actual value for period t
$\widehat{Y}_t$ : Predicted value for period t
$n$ : Lots of data

## 3. Results and Discussion

### 3.1. Descriptive Statistics

The research data used is monthly data on the number of domestic flight passengers at Kualanamu Airport from January 2019 to December 2024. The data was obtained from the Central Bureau of Statistics of the Republic of Indonesia. The passenger number data is arranged in the form of a monthly time series of 71 data points. This analysis aims to provide an overview of the data distribution, such as the average, median, minimum, maximum values, and measures of data dispersion.

**Table 1.** Airline Passenger Data for 2019-2024

| Year | January | February | March | April | May | June |
|---|---|---|---|---|---|---|
| 2019 | 280839 | 198871 | 202298 | 195395 | 155731 | 245820 |
| 2020 | 288819 | 227602 | 172348 | 29386 | 3593 | 28567 |
| : | : | : | : | : | : | : |
| 2023 | 255263 | 190693 | 197738 | 209681 | 251463 | 216748 |
| 2024 | 225772 | 178374 | 158265 | 230344 | 191394 | 190287 |

| Year | July | August | September | October | November | December |
|---|---|---|---|---|---|---|
| 2019 | 236009 | 225363 | 214364 | 225484 | 229847 | 220846 |
| 2020 | 62904 | 89451 | 74638 | 85663 | 116956 | 133574 |
| : | : | : | : | : | : | : |
| 2023 | 248679 | 209408 | 190778 | 197989 | 187237 | 187584 |
| 2024 | 216423 | 190270 | 205372 | 177282 | 167498 | |

**Table 2.** Descriptive Statistics

| Year | N | Mean | Median | Min | Max | Range | Varians | Std. Deviasi |
|---|---|---|---|---|---|---|---|---|
| 2019 | 12 | 219.238,92 | 223.104,5 | 155.731 | 280.839 | 125.108 | 936.488.991,4 | 30.602,10763 |
| 2020 | 12 | 109.458,42 | 87.557 | 3.593 | 288.819 | 285.226 | 7.202.708.481 | 343.684,876 |
| : | : | : | : | : | : | : | : | : |
| 2023 | 12 | 211.938.42 | 203.698,5 | 187.237 | 255.263 | 68.026 | 665.670.938 | 25.800,60 |
| 2024 | 11 | 193.752,82 | 190.287 | 158.265 | 230.344 | 72.079 | 549.928.768 | 23.450,56 |

Table 2 shows that the average number of air passengers per month fluctuated from 2019 to 2024. Overall, these descriptive statistics illustrate that, after experiencing a sharp decline in 2020, air passenger numbers have begun to gradually recover. The emerging pattern indicates an increasingly stable and consistent trend from year to year.

### 3.2. Transformation and Pre-Processing

#### 3.2.1. Data Normalization

Before being used in the GRU network training process, the data is first transformed into the [0,1] scale form through a normalization process. Normalization is carried out using the min-max normalization formula in Equation (5).

$$z' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

Based on passenger data from January 2020 to November 2024, 71 data were obtained:

$x_{\min} = 3.593$ (May 2020)

$x_{\max} = 288.819$ (January 2020)

$x_{\max} - x_{\min} = 285.226$

The following is a manual calculation for normalizing some data:

1. January 2019

   $x = 280839$

   $z = \frac{280839-3593}{285226}$

   $z = \frac{277246}{285226}$

   $z = 0.972022$

2. February 2019

   $x = 198871$

   $z = \frac{198871-3593}{285226}$

   $z = \frac{195278}{285226}$

   $z = 0.684643$

The following summary of the normalization results can be seen in Table 3.

**Table 3.** Passengers Number Normalization Results Data

| No | Years/Month | Number of Passengers | Normalization |
|---|---|---|---|
| 1 | 2019-01 | 280839 | 0.972022 |
| 2 | 2019-02 | 198871 | 0.684643 |

| : | : | : | : |
|---|---|---|---|
| 70 | 2024-10 | 177282 | 0.608952 |
| 71 | 2024-11 | 167498 | 0.57465 |

### 3.2.2. Sliding Window Formation and Training-Test Data Sharing

This stage aims to create pairs of input and target data that will be used for training and testing the GRU model. The formation of input and target pairs in time series data is performed using the sliding window method with a 12-month window size. This means that the first 12 months are used as input (X) and the 13th month as target/output (y). This process is carried out sequentially until the end of the data set.

$$\{x_1', x_2', x_3', \cdots, x_n'\} \text{ with } n = 59$$

So the formation of input and target pairs uses the following formula:
input: $X^{(i)}[x_1', x_2', x_3', x_4', x_5', x_6', x_7', x_8', x_9', x_{10}', x_{11}', x_{12}', \cdots, x_n']$
Here is the sliding window table that has been installed:

**Table 4.** Data *Sliding Window*

| No | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ |
|---|---|---|---|---|---|---|---|
| 1 | 0.972022 | 0.684643 | 0.696658 | 0.672456 | 0.533395 | 0.849246 | 0.814849 |
| 2 | 0.684643 | 0.696658 | 0.672456 | 0.533395 | 0.849246 | 0.814849 | 0.777524 |
| 3 | 0.696658 | 0.672456 | 0.533395 | 0.849246 | 0.814849 | 0.777524 | 0.738961 |
| : | : | : | : | : | : | : | : |
| 58 | 0,68155 | 0,64385 | 0,64507 | 0,77896 | 0,61278 | 0,54228 | 0,79499 |
| 59 | 0,64385 | 0,64507 | 0,77896 | 0,61278 | 0,54228 | 0,79499 | 0,65843 |

| No | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ | Target |
|---|---|---|---|---|---|---|
| 1 | 0.777524 | 0.738961 | 0.777948 | 0.793245 | 0.761687 | 1 |
| 2 | 0.738961 | 0.777948 | 0.793245 | 0.761687 | 1 | 0.785374 |
| 3 | 0.777948 | 0.793245 | 0.761687 | 1 | 0.785374 | 0.591654 |
| : | : | : | : | : | : | : |
| 58 | 0,658429 | 0,654548 | 0,74618 | 0,654488 | 0,707436 | 0,608952 |
| 59 | 0,654548 | 0,74618 | 0,654488 | 0,707436 | 0,608952 | 0,57465 |

From table 4, 59 pairs of input-target data were obtained. Of these, 80% or 47 pairs were used as training data, covering the period from January 2019 to January 2024. While the remaining 20% or 12 pairs were used as test data, covering the period from February 2024 to November 2024. The following is a table of training data and test data: From table 4, 59 pairs of input-target data were obtained. Of these, 80% or 47 pairs were used as training data, covering the period from January 2019 to January 2024. While the remaining 20% or 12 pairs were used as test data, covering the period from February 2024 to November 2024. The following is a table of training data and test data:

**Table 5.** Training Data

| No | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ |
|---|---|---|---|---|---|---|---|
| 1 | 0.972022 | 0.684643 | 0.696658 | 0.672456 | 0.533395 | 0.849246 | 0.814849 |
| 2 | 0.684643 | 0.696658 | 0.672456 | 0.533395 | 0.849246 | 0.814849 | 0.777524 |
| 3 | 0.696658 | 0.672456 | 0.533395 | 0.849246 | 0.814849 | 0.777524 | 0.738961 |
| : | : | : | : | : | : | : | : |
| 46 | 0,6516 | 0,64863 | 0,68418 | 0,88235 | 0,65597 | 0,68067 | 0,72254 |
| 47 | 0,64863 | 0,68418 | 0,88235 | 0,65597 | 0,68067 | 0,72254 | 0,86903 |

| No | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ | Target |
|---|---|---|---|---|---|---|
| 1 | 0.777524 | 0.738961 | 0.777948 | 0.793245 | 0.761687 | 1 |
| 2 | 0.738961 | 0.777948 | 0.793245 | 0.761687 | 1 | 0.785374 |
| 3 | 0.777948 | 0.793245 | 0.761687 | 1 | 0.785374 | 0.591654 |
| : | : | : | : | : | : | : |
| 46 | 0,86903 | 0,74732 | 0,859269 | 0,721586 | 0,656269 | 0,681551 |
| 47 | 0,74732 | 0,859269 | 0,721586 | 0,656269 | 0,681551 | 0,643854 |

Based on Table 4.5, this training data has been normalized using min-max scaling so that its values are in the range 0–1. Each row contains a sequence of passenger data using a sliding window method, so the values in the next row are shifted one step from the previous row.

**Table 6.** Test Data

| No | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ |
|---|---|---|---|---|---|---|---|
| 1 | 0.684180 | 0.882353 | 0.655971 | 0.680671 | 0.722543 | 0.869030 | 0.747320 |
| 2 | 0.882353 | 0.655971 | 0.680671 | 0.722543 | 0.869030 | 0.747320 | 0.859269 |
| 3 | 0.655971 | 0.680671 | 0.722543 | 0.869030 | 0.747320 | 0.859269 | 0.721586 |
| : | : | : | : | : | : | : | : |
| 11 | 0.681551 | 0.643854 | 0.645071 | 0.778958 | 0.612781 | 0.542279 | 0.794987 |
| 12 | 0.643854 | 0.645071 | 0.778958 | 0.612781 | 0.542279 | 0.794987 | 0.658429 |

| No | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ | Target ($y$) |
|---|---|---|---|---|---|---|
| 1 | 0.859269 | 0.721586 | 0.656269 | 0.681551 | 0.643854 | 0.645071 |
| 2 | 0.721586 | 0.656269 | 0.681551 | 0.643854 | 0.645071 | 0.778958 |
| 3 | 0.656269 | 0.681551 | 0.643854 | 0.645071 | 0.778958 | 0.612781 |
| : | : | : | : | : | : | : |
| 11 | 0.658429 | 0.654548 | 0.746180 | 0.654488 | 0.707436 | 0.608952 |
| 12 | 0.654548 | 0.746180 | 0.654488 | 0.707436 | 0.608952 | 0.574650 |

Based on Table 6, this test data has also been normalized to the 0–1 range with min-max scaling and arranged using the sliding window method, just like the training data. The difference is that the test data is used to measure the model's performance on data it has never seen during training.

### 3.3. Hyperparameter Initialization

The determination of hyperparameter values has been determined by researchers as follows:

**Table 7.** GRU Model Hyperparameter Values

| Parameter Name | Value | Description |
|---|---|---|
| *Window size* | 12 | Time series input window length (last 12 months as input) |
| Jumlah Fitur per *Timestep* | 2 | Normalized passenger count & seasonal features (month) |
| *Hidden layer* 1 (*GRU*) | 256 *unit*, `return_sequences=True` | Saves sequence for input to next GRU |
| *Dropout* setelah *GRU* 1 | 0.2 | Regulatization to prevent overfitting |
| *Hidden layer* 2 (*GRU*) | 128 *unit* | Second GRU layer without return sequences |

| | | |
|---|---|---|
| *Dropout* setelah *GRU* 2 | 0.2 | *Additional dropout for learning stabilization* |
| *Hidden layer* (*Dense)* | 64 *unit*, *ReLU activation* | Fully connected non-linear layer before output |
| *Output layer* | 1 *unit*, *linear* | Prediction of the next number of passengers |
| *Optimizer* | *Adam* | *Stable adaptive-based optimizer* |
| *Learning rate* | 0.0001 | Low learning rate for stability |
| *Loss function* | *Mean Squared Error* (*MSE*) | Suitable for time series regression |
| *Batch size* | 8 | Number of data in one iteration of weight update |
| *Enochs* | 200 | Number of training iterations on all data |
| Validasi | 20% *test* data | Validation during training using test data |
| *Model checkpoint* | Save the best `val_loss` | The model is saved automatically when the best validation result is achieved |

**Table 8.** Models Formed by Python

Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| gru (GRU) | (None, 12, 256) | 199,680 |
| dropout (Dropout) | (None, 12, 256) | 0 |
| gru_1 (GRU) | (None, 128) | 148,224 |
| dropout_1 (Dropout) | (None, 128) | 0 |
| dense (Dense) | (None, 64) | 8,256 |
| dense_1 (Dense) | (None, 1) | 65 |

Total params: 1,068,677 (4.08 MB)
Trainable params: 356,225 (1.36 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 712,452 (2.72 MB)

From Table 8, the GRU model consists of six main layers: two GRU layers, two dropout layers, and two dense layers. Overall, the model has a total of 1,068,677 parameters, consisting of 356,225 trainable parameters and 712,452 parameters used by the optimizer. There are no non-trainable parameters in this model. From Table 8, the GRU model consists of six main layers: two GRU layers, two dropout layers, and two dense layers. Overall, the model has a total of 1,068,677 parameters, consisting of 356,225 trainable parameters and 712,452 parameters used by the optimizer. There are no non-trainable parameters in this model.

### 3.4. GRU Training Process

The GRU model training process was performed using previously prepared training and validation data in a sliding window format. The model was trained for 200 epochs, with a batch size of 8.

Epoch 1/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **5s** 166ms/step - loss: 0.2290 - val_loss: 0.2312
Epoch 2/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **1s** 55ms/step - loss: 0.1551 - val_loss: 0.1102
Epoch 3/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **1s** 55ms/step - loss: 0.1024 - val_loss: 0.0359

Epoch 4/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **1s** 97ms/step - loss: 0.0429 - val_loss: 0.0130
Epoch 5/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **1s** 99ms/step - loss: 0.0400 - val_loss: 0.0127
Epoch 6/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **1s** 96ms/step - loss: 0.0447 - val_loss: 0.0119
Epoch 7/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **0s** 84ms/step - loss: 0.0505 - val_loss: 0.0130
Epoch 8/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **0s** 45ms/step - loss: 0.0340 - val_loss: 0.0166
Epoch 9/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **0s** 44ms/step - loss: 0.0450 - val_loss: 0.0178
Epoch 10/200
**6/6** ━━━━━━━━━━━━━━━━━━━━ **0s** 47ms/step - loss: 0.0402 - val_loss: 0.0172

Based on the results of 200 epochs of GRU model training, the model demonstrated quite good performance in learning patterns in passenger time series data, taking seasonality into account. This indicates that the model managed to maintain consistent performance until the end of training, although performance improvement was no longer significant in the final phase. Overall, the model's performance was deemed stable and optimal in capturing seasonal patterns and historical trends in the passenger data used.

### 3.5. GRU Model Testing

The testing process is carried out by inputting test data into the GRU model. The model output results in the form of predicted values on a normalized scale as follows:

**Table 9.** Normalization Prediction Results

| No | Actual (Scaled) | Prediction (Scaled) |
|---|---|---|
| 1 | 0.645071 | 0.711380 |
| 2 | 0.778958 | 0.731915 |
| 3 | 0.612781 | 0.616444 |
| : | : | : |
| 11 | 0.608952 | 0.715832 |
| 12 | 0.574650 | 0.693545 |

Based on Table 9, it can be seen that most of the predicted values have a relatively small difference from the actual values, indicating that the model is quite capable of following the data pattern. However, at some points, such as row 5, the difference between the actual and predicted values appears larger, indicating a prediction error that needs attention.

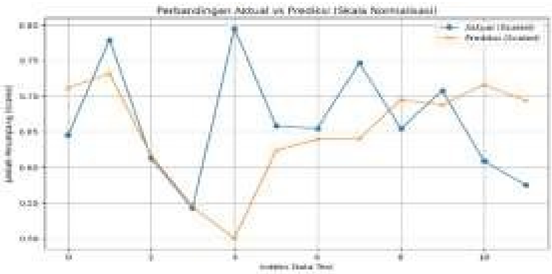

**Figure 1.** Normalized Prediction Graph

Based on the figure, the line on the graph shows that most of the predicted points are close to the actual values, so the actual data pattern can be followed well. Next, these predicted values are returned to their original scale using the Min-Max denormalization method. The denormalization results are compared with the actual data in the testing period. The following table shows the denormalization prediction results:

**Table 10.** Denormalization Prediction Results

| No | Actual | Prediction |
|---|---|---|
| 1 | 187584 | 206497 |
| 2 | 225772 | 212354 |
| 3 | 178374 | 179419 |
| : | : | : |
| 11 | 177282 | 207767 |
| 12 | 167498 | 201410 |

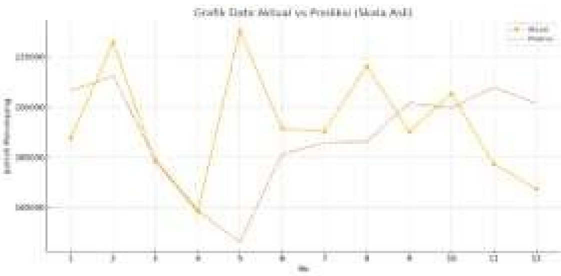

**Figure 2.** Normalized Prediction Graph

Figure 2 shows that the GRU model can follow the general pattern and trend of the actual data during the 12 months of testing quite well, although there are deviations in some periods. In months 1 to 3, the predictions are very close to the actual values, while in months 4 and 5 there are overestimates, and in month 8 there are underestimates. However, these differences are not too extreme, and the model still captures the trend direction and seasonal fluctuations. Overall, the GRU performance is considered quite good and suitable for use in forecasting similar data.

### 3.6. Model Evaluation

This evaluation process uses two common metrics, namely Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE).

**Table 11.** Model Evaluation

| No | Actual ($Y_t$) | Prediction ($\widehat{Y_t}$) | Difference ($\widehat{Y_t} - Y_t$) | $(\widehat{Y_t} - Y_t)^2$ | RMSE | MAPE (%) |
|---|---|---|---|---|---|---|
| 1 | 187.584 | 206.497 | 18.913 | 357.701.569 | 30.134,99 | 10,0824 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 225.772 | 212.354 | -13.418 | 180.042.724 | | 5,9432 |
| : | : | : | : | : | : | : |
| 11 | 177.282 | 207.767 | 30.485 | 929.335.225 | | 17,1958 |
| 12 | 167.498 | 201.410 | 33.912 | 1.150.023.744 | | 20,2462 |
| Amount | 2.318.865 | 2.267.374 | -51.491 | 10.897.411.581 | | 121,2147 |
| Average | 193.238,75 | 188.947,83 | -4.290,92 | 908.117.631,8 | | 10,1012 |

The proposed GRU model was trained and tested using the prepared dataset. The evaluation results show that the model achieved a Mean Absolute Percentage Error (MAPE) of 10.10% and a Root Mean Square Error (RMSE) of 30,134.99. According to common forecasting criteria, a MAPE value below 20% indicates good forecasting accuracy. Therefore, these results confirm that the GRU model can produce reliable and accurate predictions for air passenger data.

### 3.7. Passenger Forecast for December 2024-November 2026

The prediction results demonstrate that the GRU model can closely follow the actual data patterns. The model successfully captures both increasing and decreasing trends, as well as seasonal fluctuations. The forecasting results for the next 24 months show a consistent seasonal pattern, where passenger numbers tend to increase during the beginning and end of the year and decrease in the middle of the year. This pattern is consistent with real-world conditions, particularly during major holiday periods

**Table 12.** Forecast Results for 24 months

| No | Year-Month | Passenger Number Prediction |
|---|---|---|
| 1 | 2024-12 | 196079 |
| 2 | 2025-01 | 213325 |
| 3 | 2025-02 | 176507 |
| | | |
| 23 | 2026-10 | 175334 |
| 24 | 2026-11 | 190169 |

The table above shows a fluctuation pattern reflecting seasonal characteristics, with passenger numbers tending to increase in certain months, such as December–January, and decrease mid-year, particularly in May–July. This aligns with historical trends, which are typically influenced by year-end holidays, the homecoming season, and low travel periods.

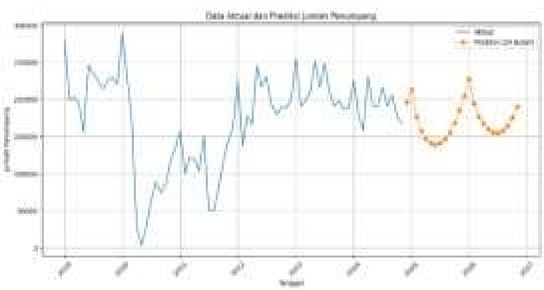

**Figure 3.** Actual and Predicted Data Graphs

Figure 3, which displays a graph of actual and predicted data, shows that the predicted line follows the historical data pattern quite well. The peaks and valleys in the predicted graph align with the previous actual data, indicating that the model successfully captures both seasonal patterns and long-term trends.

### 3.8. Discussion and Novelty

The findings of this study provide several important insights. First, the GRU model shows strong capability in modeling nonlinear and seasonal patterns in time series data. Compared to traditional statistical methods such as ARIMA and exponential smoothing, the GRU model demonstrates better flexibility and accuracy in capturing complex relationships.

Second, the use of a multi-layer GRU architecture improves the model's ability to learn temporal dependencies. Unlike single-layer models commonly used in previous studies, the multi-layer structure enables the extraction of deeper features, resulting in improved forecasting performance. Third, the integration of seasonal features significantly enhances the model's ability to recognize recurring patterns in the data.

This is particularly relevant for air passenger data, which are strongly influenced by seasonal events. These aspects represent the main novelty of this study. Specifically, this research introduces a multi-layer GRU model combined with seasonal feature inputs, which has not been widely explored in previous studies on air passenger forecasting. However, some prediction errors are still observed in certain periods. These errors may be caused by external factors that are not included in the model, such as economic conditions, policy changes, or unexpected events. Therefore, future studies are recommended to incorporate additional external variables to further improve model performance.

Overall, the proposed model provides a robust and effective approach for forecasting air passenger numbers and offers practical benefits for airport capacity planning and operational decision-making.

## 4. Conclusion

This study successfully developed a forecasting model for domestic air passenger numbers at Kualanamu International Airport using the Gated Recurrent Unit (GRU) method. The results show that the model achieves good forecasting performance with a MAPE value of 10.10% and RMSE of 30,134.99.

The forecasting results for the next 24 months indicate consistent seasonal patterns, with peak passenger numbers occurring at the beginning and end of the year, and lower values in the middle of the year. The main contribution of this study lies in the application of a multi-layer GRU architecture combined with seasonal features, which improves the model's ability to capture complex patterns in time series data.

## References

[1] S. Sofiana, S. Suparti, A. R. Hakim, and I. Triutami, "Forecasting the number of airplane passengers at Ahmad Yani International Airport using the Holt-Winters exponential smoothing method and event-based exponential smoothing", *Jurnal Gaussian*, vol. 9, no. 4, pp. 535-545, 2020.

[2] H. Silitonga, I. Gunawan, and B. E. Damanik, "The artificial neural network (ANN) algorithm implementation for predicting book sales at Erlangga Publisher Pematang Siantar", *International Journal of Multi Science*, vol. 1, no. 12, pp. 1-8, 2021.

[3] L. Wiranda and M. Sadikin, "Application of long short-term memory for time series data to predict product sales at PT Metiska Farma", *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 8, no. 3, pp. 184-196, 2019.

[4] D. Wira, *Fundamental stock analysis*, 2nd ed., Exceed Books, 2014.

[5] R. A. Saputra, Y. Azhar, and V. Rahmayanti, "Cargo demand prediction at Tangerang

City Cargo Service Center using the Gated Recurrent Unit method”, *Jurnal Repositor*, vol. 2, no. 8, 2020.

[6] A. Aulia, A. M. Elhanafi, and H. Dafitri, "Implementation of the Gated Recurrent Unit algorithm for predicting palm oil prices using recurrent neural networks", *Proceedings of SNASTIKOM*, pp. 288–294, 2021.

[7] N. H. Judoprajitno, "McDonald's stock price prediction on the New York Stock Exchange using the Gated Recurrent Unit method", *Undergraduate thesis*, Universitas Atma Jaya Yogyakarta, 2021.

[8] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and practice*, 2nd ed. OTexts, 2018. [Online]. Available: https://otexts.org/fpp2/

[9] Z. Latif, "Decision support system for determining the most favorite airline using the simple additive weighting method", *Undergraduate thesis*, Institut Teknologi Dirgantara Adisutjipto, 2023.

[10] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks", *arXiv preprint arXiv:1801.01078*, 2017.

[11] W. S. Nugroho, I. Fitri, A. Surya, and S. Suwanto, "Application of deep learning using Gated Recurrent Unit to predict world crude oil prices", *Building Informatics, Technology and Science*, vol. 5, no. 1, pp. 86–94, 2023.

[12] M. Atharsyah and M. A. Romli, "Implementation of LSTM, GRU, BiLSTM, and BiGRU models in nickel price prediction", *Jurnal Informatika dan Rekayasa Elektronik*, vol. 7, no. 2, pp. 413–420, 2024.

[13] Y. Abdillah and S. Suharjito, "Failure prediction of e-banking application systems using adaptive neuro fuzzy inference system (ANFIS)", *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, p. 667, 2019.

[14] K. Khoiri, "How to calculate mean absolute percentage error (MAPE)", 2020. [Online]. Available: https://www.khoiri.com