

## PREDIKSI KEBUTUHAN AIR UNTUK PENYIRAMAN TANAMAN PAPRIKA PADA SISTEM PERTANIAN *GREENHOUSE* MENGGUNAKAN METODE *GATE RECURRENT UNIT (GRU)*

Sofiyatus Zawayah<sup>1</sup>, Miftahul Walid<sup>1</sup>, Fathorrozi Ariyanto<sup>1</sup>

<sup>1</sup>Teknik Informatika, Teknik, Universitas Islam Madura

<sup>1</sup>sfiyatszwyh@gmail.com, <sup>2</sup>miftahwalid@gmail.com\*, <sup>3</sup>fathorroziaryanto7@gmail.com

### ABSTRAK

Pertanian berkelanjutan menuntut efisiensi penggunaan sumber daya, terutama air, yang sering kali menjadi faktor pembatas dalam budidaya tanaman, termasuk di sistem *greenhouse*. Penelitian ini berfokus pada pengembangan model prediksi kebutuhan air tanaman paprika menggunakan algoritma *deep learning Gated Recurrent Unit (GRU)*. Pemilihan *GRU* didasarkan pada kemampuannya dalam mengolah data deret waktu yang bersifat kompleks serta dinamis. Data sekunder yang digunakan diperoleh dari repositori *4TU.ResearchData*, mencakup parameter lingkungan seperti suhu, kelembapan, dan radiasi, dengan volume irigasi sebagai target prediksi. Proses penelitian meliputi normalisasi data, pembagian data latih dan uji, pelatihan model, denormalisasi hasil prediksi, serta evaluasi performa dengan metrik *MAE*, *MSE*, *RMSE*, dan  $R^2$ . Hasil pengujian menunjukkan bahwa *GRU* mampu menghasilkan prediksi dengan tingkat akurasi sangat tinggi, ditunjukkan oleh nilai  $R^2$  sebesar 0,9979, *MAE* sebesar 6,88, *MSE* sebesar 140,87, dan *RMSE* sebesar 11,87. Temuan ini membuktikan bahwa model *GRU* efektif dalam memprediksi kebutuhan air secara akurat dan dapat diimplementasikan dalam sistem irigasi otomatis di *greenhouse*. Dengan demikian, model ini berpotensi mendukung efisiensi penggunaan air dan peningkatan produktivitas pertanian berbasis teknologi cerdas.

**Kata kunci:** *GRU*, efisiensi air, *greenhouse*, kecerdasan buatan, pertanian berkelanjutan.

### ABSTRACT

*Sustainable agriculture requires efficient resource utilization, particularly water, which often becomes a limiting factor in crop cultivation, including greenhouse systems. This study focuses on developing a prediction model for the water requirements of paprika plants using the Gated Recurrent Unit (GRU) deep learning algorithm. The GRU was selected due to its ability to process complex and dynamic time-series data. The secondary data were obtained from the 4TU.ResearchData repository, covering environmental parameters such as temperature, humidity, and radiation, with irrigation volume as the prediction target. The research process involved data normalization, splitting into training and testing sets, model training, prediction denormalization, and performance evaluation using MAE, MSE, RMSE, and R<sup>2</sup> metrics. The evaluation results indicate that GRU is capable of generating highly accurate predictions, demonstrated by an R<sup>2</sup> value of 0.9979, MAE of 6.88, MSE of 140.87, and RMSE of 11.87. These findings confirm that the GRU model is effective in accurately predicting water demand and can be implemented in automatic greenhouse irrigation systems. Thus, this model has the potential to support efficient water usage and enhance agricultural productivity through smart technology.*

**Keywords:** *GRU*, water efficiency, *greenhouse*, artificial intelligence, sustainable agriculture.

### 1. PENDAHULUAN

Pertanian merupakan sektor vital yang menopang kebutuhan pangan masyarakat di

seluruh dunia [1]. Dalam perkembangannya, sektor pertanian menghadapi tantangan dalam meningkatkan produktivitas di tengah keterbatasan sumber daya, terutama air. Pada

praktiknya, penggunaan air dalam pertanian sering tidak efisien, sehingga menimbulkan pemborosan dan meningkatkan biaya operasional [2]. Masalah ini menjadi semakin krusial mengingat banyak daerah yang menghadapi kelangkaan air. Upaya untuk mencapai pertanian berkelanjutan menuntut penerapan teknologi yang mendukung efisiensi penggunaan air dalam proses budidaya tanaman [3].

Salah satu inovasi dalam pertanian modern yang mendukung penggunaan air yang lebih efisien adalah sistem *greenhouse* atau rumah kaca. *Greenhouse* memungkinkan pengelolaan lingkungan tumbuh secara lebih terkendali, seperti suhu, kelembaban, dan intensitas cahaya, sehingga dapat mengoptimalkan pertumbuhan tanaman sepanjang tahun [4]. Namun, penggunaan air dalam *greenhouse* masih memerlukan pendekatan yang tepat untuk memastikan bahwa kebutuhan tanaman terpenuhi secara akurat, tanpa pemborosan. Dalam konteks ini, sistem prediksi kebutuhan air dapat menjadi solusi untuk meningkatkan efisiensi penyiraman tanaman [5]

Masalah utama dalam penyiraman di *greenhouse* adalah ketidakpastian kebutuhan air tanaman yang bervariasi berdasarkan faktor lingkungan dan fase pertumbuhan tanaman [6]. Kebutuhan air yang tidak teratur atau berlebihan dapat mengganggu pertumbuhan dan kualitas hasil panen. Penyiraman yang terlalu sering atau terlalu jarang akan mempengaruhi kadar air dalam tanah, yang pada gilirannya dapat mempengaruhi produktivitas tanaman [7]. Dengan demikian, dibutuhkan suatu metode yang dapat memperkirakan kebutuhan air secara lebih tepat berdasarkan kondisi tanaman dan faktor lingkungan dalam *greenhouse*.

Beberapa penelitian terdahulu tentang prediksi kebutuhan air telah dikembangkan, antara lain; Penelitian oleh Michael et al. [8] sistem irigasi cerdas berbasis *Adaptive Neuro-Fuzzy Inference System (ANFIS)* untuk memprediksi kebutuhan penyiraman tanaman cabai secara presisi. Penelitian oleh D. Maria Manuel Vianny et al. [9] mengemukakan bahwa penerapan sistem irigasi presisi berbasis *Internet of Things (IoT)* dan algoritma pembelajaran mesin, seperti *K-Nearest Neighbors (KNN)*, *Gradient Boosting-based*

*Tree (GBT)*, dan *Long Short-Term Memory (LSTM)*, dapat secara signifikan meningkatkan efisiensi penggunaan air dalam pertanian. Penelitian oleh Kurniasari et al. [10] mengembangkan model peramalan nilai impor minyak, non-minyak, dan gas Indonesia menggunakan metode *Gated Recurrent Unit (GRU)*, yang menunjukkan akurasi tinggi sebesar 99.000044% dalam memprediksi fluktuasi nilai impor dari Januari 2000 hingga Juni 2022. Penelitian sebelumnya oleh He et al. [11] mengembangkan metode prediksi aliran harian menggunakan pendekatan *Gated Recurrent Unit (GRU)* yang berbasis pada dekomposisi musiman, yang menunjukkan efektivitas dalam memproses data deret waktu non-linear.

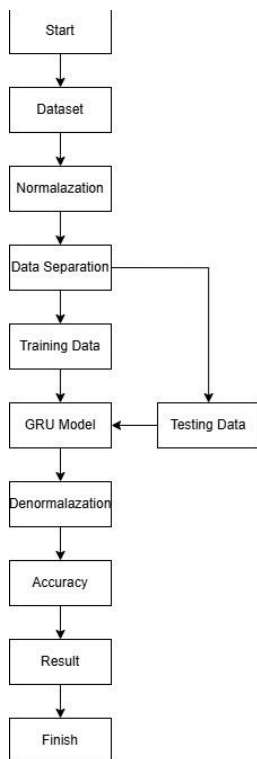
Namun, penelitian diatas belum ada yang menggunakan model *deeplearning* khususnya *Gate Recurrent Unit (GRU)* pada sistem pertanian *greenhouse*. Dengan demikian, studi ini dimaksudkan untuk merancang sebuah model *deeplearning* menggunakan metode *GRU* untuk memprediksi kebutuhan air dalam penyiraman tanaman paprika pada sistem pertanian *greenhouse*. *Gate Recurrent Unit (GRU)* merupakan salah satu teknik dalam jaringan saraf tiruan yang memiliki keunggulan dalam memproses data deret waktu [12]. *GRU* dirancang untuk menangkap pola data dengan ketergantungan waktu, sehingga cocok digunakan untuk memprediksi kebutuhan air tanaman yang dipengaruhi oleh perubahan faktor lingkungan dari waktu ke waktu [13].

Dengan menggunakan model prediksi berbasis *GRU*, Studi ini difokuskan pada pengembangan model yang memiliki kemampuan untuk memprediksi kebutuhan air secara akurat bagi tanaman paprika di *greenhouse*. Model yang dikembangkan diharapkan mampu mengoptimalkan efisiensi penyiraman tanaman, mengurangi pemborosan air, serta berkontribusi besar terhadap implementasi pertanian berkelanjutan yang ditunjang oleh teknologi cerdas [14].

## 2. METODOLOGI PENELITIAN

Pendekatan yang digunakan dalam penelitian ini adalah kuantitatif, dengan algoritma *Gate Recurrent Unit (GRU)* untuk memprediksi kebutuhan air tanaman paprika

dalam sistem pertanian greenhouse. Proses pemodelan dilakukan menggunakan perangkat lunak berbasis *Python* yaitu *google colab* dengan pustaka seperti *TensorFlow* atau *PyTorch* untuk implementasi model *GRU*. Berikut merupakan urutan tahapan yang diterapkan dalam penelitian ini:



Gambar 1. Tahapan Penelitian

Berdasarkan *flowchart* yang digambarkan pada Gambar 1, proses penelitian dilakukan melalui tahapan berikut:

### 2.1 Dataset

Dataset penelitian ini memanfaatkan data sekunder yang diambil dari repositori *4TU.ResearchData* dengan kata kunci "*Greenhouse 2016*", serta didukung oleh riset yang membahas penerapan model emisi *greenhouse* untuk perlindungan tanaman dengan irigasi tetes pada media tanam tanpa tanah.

Dataset dapat diakses melalui: <https://data.4tu.nl/search?search=Greenhouse+2016>. Dataset ini terdiri dari 2510 sampel yang merepresentasikan kondisi lingkungan dalam *greenhouse*. Meskipun dataset memiliki

sejumlah variabel, penelitian ini hanya memanfaatkan empat variabel, yaitu radiasi (*radiation*), suhu (*temperature*), dan kelembapan (*humidity*) sebagai variabel input, serta volume irigasi (*volume of irrigation*) sebagai variabel output. Seluruh data diunduh dari *4TU.ResearchData* kemudian dieksplorasi untuk memastikan kelengkapan serta kualitasnya sebelum diterapkan dalam tahap training model.

### 2.2 Normalization

Sebelum data dapat digunakan dalam training model, data terlebih dahulu diproses pada tahap *preprocessing*, salah satunya adalah normalisasi data. Pada tahap ini, setiap fitur distandarkan ke dalam skala [0,1] dengan pendekatan *Min-Max Scaling*. Tujuan dari normalisasi ini adalah untuk memastikan distribusi data lebih seragam, sehingga tahapan training model menjadi lebih optimal dan konsisten. Normalisasi dilakukan dengan rumus:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Di mana:

$X$ : Nilai asli dari data sebelum melalui proses normalisasi

$X_{min}$ : Nilai terkecil dari semua data dalam kolom atau fitur tertentu

$X_{max}$ : Nilai terbesar dari seluruh data dalam kolom atau fitur tersebut

$X_{scaled}$ : Nilai data setelah dinormalisasi ke rentang 0 sampai 1

### 2.3 Data Separation

Pada bagian ini, dataset terbagi menjadi dua subset yakni data training dan data testing. Dari total dataset, 80% bagian dimanfaatkan sebagai data latih yang berfungsi untuk membangun dan mengoptimalkan model sehingga mampu mengidentifikasi pola dalam data. Sementara itu, 20% sisanya digunakan sebagai data uji yang digunakan dalam proses evaluasi kinerja model terhadap data yang sama sekali belum dipelajari pada tahap training. Tujuan pemisahan ini adalah mengevaluasi sejauh mana model mampu melakukan generalisasi secara obyektif pada data yang belum dikenalnya.

## 2.4 Training Data

Selama proses pelatihan, data latih digunakan untuk menyesuaikan bobot-bobot dalam arsitektur *GRU* berdasarkan pola yang ditemukan dalam fitur input. Model dilatih dalam beberapa epoch dengan menggunakan fungsi *loss* dan optimizer tertentu untuk meminimalkan kesalahan prediksi. Parameter yang digunakan mencakup *learning rate*, *batch size*, dan jumlah unit *GRU* disesuaikan melalui proses eksperimen guna memperoleh output yang optimal. Maksud dari pelatihan ini dirancang guna membentuk model *GRU* yang mampu memahami pola historis pada data, sehingga mampu menghasilkan prediksi dengan tingkat akurasi yang tinggi ketika diterapkan pada data uji.

## 2.5 GRU Model

Arsitektur model *GRU* dirancang untuk menangani data deret waktu dengan mempertimbangkan tiga gerbang utama:

*Reset Gate* ( $r_t$ ): Mengatur informasi yang perlu dilupakan dari *hidden state* sebelumnya.

*Update Gate* ( $z_t$ ): Menetapkan tingkat seberapa besar *hidden state* sebelumnya dipertahankan atau diperbarui.

*Candidate Hidden State* ( $\tilde{h}_t$ ): Menghasilkan *hidden state* baru berdasarkan informasi yang tersedia.

Persamaan matematis yang digunakan dalam *GRU* adalah sebagai berikut:

### 2.5.1 Reset Gate ( $r_t$ ):

$$r_t = \sigma(W_r \cdot [h_t - 1, x_t] + b_r) \quad (1)$$

Di mana:

$r_t$ : Reset gate pada waktu

$\sigma$ : fungsi aktivasi sigmoid, yang menghasilkan output pada rentang 0 hingga 1

$W_r$ : bobot reset gate

$h_t - 1, x_t$ : penggabungan antara input dan *hidden state* sebelumnya

$h_t - 1$ : *hidden state* pada waktu sebelumnya

$b_r$ : bias reset gate

### 2.5.2 Update Gate ( $z_t$ ):

$$z_t = \sigma(W_z \cdot [h_t - 1, x_t] + b_z) \quad (2)$$

Di mana:

$z_t$ : update gate, nilai antara 0 dan 1 yang menentukan seberapa banyak informasi baru yang akan digunakan.

$\sigma$ : Fungsi aktivasi sigmoid, yang membatasi nilai output antara 0 dan 1.

$W_z$ : Matriks bobot untuk update gate — belajar dari data selama proses pelatihan.

$h_t - 1, x_t$ : Penggabungan vektor *hidden state* sebelumnya ( $h_t - 1$ ) dan input saat ini ( $x_t$ ). Biasanya dilakukan concatenation.

$b_z$ : Bias untuk update gate.

### 2.5.3 Candidate Hidden State ( $\tilde{h}_t$ ):

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_t - 1, z_t] + b_h) \quad (3)$$

Di mana:

$\tilde{h}_t$ : Candidate *hidden state* pada waktu  $t$ , ini adalah calon nilai baru dari *hidden state*.

$\tanh$ : Fungsi aktivasi hyperbolic tangent; membatasi nilai output antara -1 dan 1.

$W_h$ : Matriks bobot untuk menghasilkan  $\tilde{h}_t$ :

$r_t \odot h_t - 1, z_t$ : Penggabungan dua vektor (concatenation):

$r_t \odot h_t - 1$ : hasil perkalian elemen-wise antara reset gate dan *hidden state* sebelumnya.

$z_t$ : update gate, nilai antara 0 dan 1 yang menentukan seberapa banyak informasi baru yang akan digunakan.

$b_h$ : bias untuk candidate *hidden state*.

$\odot$ : elemen-wise multiplication (perkalian per elemen antara dua vektor).

### 2.5.4 Final Hidden State ( $h_t$ ):

$$h_t = (1 - z_t) \odot h_t - 1 + z_t \odot \tilde{h}_t \quad (4)$$

Di mana:

$h_t$ : *hidden state* pada waktu, yaitu output utama *GRU* pada waktu tersebut.

$h_t - 1$ : *hidden state* pada waktu sebelumnya.

$\tilde{h}_t$ : candidate *hidden state*, yaitu hasil perhitungan dari informasi baru.

$z_t$ : update gate, nilai antara 0 dan 1 yang menentukan seberapa banyak informasi baru yang akan digunakan.

$\odot$ : elemen-wise multiplication (perkalian per elemen antara dua vektor).

## 2.6 Testing Data

Setelah tahap pelatihan selesai, model diuji dengan data uji untuk menilai kinerjanya pada data yang belum pernah diproses sebelumnya. Model diuji pada 20% data uji untuk mengevaluasi kemampuan model dalam melakukan generalisasi terhadap data baru. Hasil prediksi kemudian dibandingkan dengan

label sebenarnya menggunakan metrik evaluasi seperti akurasi atau *MSE*.

## 2.7 Denormalization

Setelah model menghasilkan prediksi, jika data target sudah dinormalisasi sebelumnya, maka tahap berikutnya adalah melakukan denormalisasi. Proses ini bertujuan Agar hasil prediksi dikembalikan ke skala awal atau rentang nilai asli, yang memungkinkan perbandingan antara prediksi model dan data asli yang berada dalam rentang yang sama dengan data yang digunakan pada tahap awal (sebelum dinormalisasi). Denormalisasi ini diperlukan untuk menjamin bahwa hasil yang diperoleh dari model dapat diinterpretasikan dengan benar sesuai dengan konteks atau satuan data yang asli.

## 2.8 Accuracy

Setelah tahap pelatihan model *GRU* selesai dilakukan, tahap berikutnya adalah mengevaluasi kinerja model dengan menggunakan beberapa metrik evaluasi, yaitu *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, dan *R-squared (R<sup>2</sup>)*. Evaluasi ini dilakukan dengan tujuan untuk menilai seberapa baik model dapat memprediksi kebutuhan air pada tanaman cabai di lingkungan *greenhouse*. Berikut rumus tersebut:

### 2.8.1 Mean Absolute Error (MAE):

*Mean Absolute Error (MAE)* berfungsi untuk menentukan rata-rata dari kesalahan absolut antara nilai asli dan output prediksi. Kesalahan absolut artinya selisih tanpa memperhatikan tanda (positif atau negatif). *Mean Absolute Error (MAE)* menggambarkan nilai rata-rata prediksi menyimpang dari nilai sebenarnya, dengan skala yang sama seperti data aslinya. Nilai *MAE* yang semakin kecil menunjukkan model semakin akurat. *MAE* juga memiliki keunggulan karena tidak terlalu sensitif terhadap *outlier*.

Rumus:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

Di mana:

*n*: Jumlah seluruh data

*y<sub>i</sub>*: Nilai aktual atau sebenarnya pada data ke-*i*

*ŷ<sub>i</sub>*: Nilai prediksi pada data ke-*i*

*y<sub>i</sub> - ŷ<sub>i</sub>*: Selisih absolut (tanpa tanda negatif) antara nilai sebenarnya dan prediksi

### 2.8.2 Mean Squared Error (MSE):

*Mean Squared Error (MSE)* menghitung nilai rata-rata dari kuadrat selisih antara data sebenarnya dan hasil prediksi. Karena menggunakan kuadrat, *MSE* akan lebih menekankan error yang signifikan, sehingga sensitif terhadap *outlier*. Metode ini sering digunakan karena mempertimbangkan besar kesalahan secara signifikan. Namun, satuannya adalah kuadrat dari satuan asli, sehingga tidak selalu intuitif untuk diinterpretasikan secara langsung.

Rumus:

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

Di mana:

*n*: Jumlah seluruh data

*y<sub>i</sub>*: Nilai aktual atau sebenarnya pada data ke-*i*

*ŷ<sub>i</sub>*: Nilai prediksi pada data ke-*i*

$(y_i - \hat{y}_i)^2$ : Kuadrat dari selisih antara nilai sebenarnya dan prediksi

### 2.8.3 Root Mean Squared Error (RMSE):

*RMSE* adalah akar dari *MSE*. Metode ini juga mengukur rata-rata kesalahan kuadrat, namun karena diakarkan, hasil akhirnya kembali ke satuan aslinya, sehingga lebih mudah diinterpretasikan dibanding *MSE*. *RMSE* tetap sensitif terhadap *outlier*, tetapi lebih intuitif untuk menilai sejauh mana prediksi model dari nilai aktual secara umum.

Rumus:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3)$$

Di mana:

*n*: Jumlah seluruh data

*y<sub>i</sub>*: Nilai aktual atau nilai sebenarnya pada data ke-*i*

*ŷ<sub>i</sub>*: Nilai prediksi pada data ke-*i*

$(y_i - \hat{y}_i)^2$ : Kuadrat dari perbedaan antara nilai asli dan nilai yang diprediksi

### 2.8.4 R-squared (R<sup>2</sup>):

*R-squared* atau koefisien determinasi merupakan metrik evaluasi yang mengukur proporsi variansi dalam variabel dependen yang dapat diprediksi oleh model. Nilai *R<sup>2</sup>* berada antara 0 dan 1. Nilai mendekati 1 menunjukkan bahwa model memiliki performa yang sangat baik dalam menjelaskan data,

sedangkan nilai dekat 0 menandakan model kurang mampu menjelaskan variabilitas data. Jika  $R^2$  negatif, itu berarti model kurang baik dibandingkan model rata-rata (*mean*).

Rumus:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4)$$

Di mana:

$n$ : Jumlah seluruh data

$y_i$ : Nilai aktual atau nilai sebenarnya pada data ke- $i$

$\hat{y}_i$ : Nilai prediksi pada data ke- $i$

$\bar{y}$ : Mean (rata-rata) dari nilai aktual  $y$

$(y_i - \hat{y}_i)^2$ : Kuadrat dari perbedaan antara nilai sebenarnya dan prediksi, yang menunjukkan kesalahan model.

$(y_i - \bar{y})^2$ : Selisih kuadrat antara nilai aktual dan rata-rata nilai aktual, yang menunjukkan variabilitas dalam data asli.

Jika hasil evaluasi menunjukkan performa yang akurat, model ini dapat digunakan untuk mendukung sistem irigasi otomatis dalam *greenhouse* guna mengoptimalkan penggunaan air bagi tanaman.

### 3. HASIL DAN PEMBAHASAN

Tujuan dari penelitian ini adalah untuk memprediksi kebutuhan irigasi tanaman paprika di dalam *greenhouse* dengan menggunakan model *Gated Recurrent Unit (GRU)*. Proses dilakukan melalui beberapa tahap, dimulai dengan pengumpulan data, *preprocessing*, pemisahan data, pelatihan model, pengujian, serta evaluasi hasil.

#### 3.1 Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini diunduh dari repositori *4TU.ResearchData* dengan kata kunci "*Greenhouse 2016*". Sebelum diterapkan pada model prediksi, dataset tersebut melewati tahap pemrosesan guna menjamin kualitas data yang optimal. Proses pemrosesan dimulai dengan tahap pembersihan data (*data cleaning*), yang mencakup pengecekan terhadap nilai yang hilang (*missing values*), duplikasi data, serta inkonsistensi dalam data. Untuk menangani nilai yang hilang, diterapkan teknik imputasi untuk mengisi nilai kosong, atau jika perlu, baris yang mengandung data yang hilang dihapus, sesuai dengan kebijakan pengolahan

data yang diterapkan. Setelah tahap pembersihan, dilakukan validasi dan verifikasi untuk memastikan bahwa dataset yang digunakan sudah tidak mengandung kesalahan serta siap diproses lebih lanjut, sehingga menghasilkan dataset yang berkualitas dan dapat diandalkan untuk analisis lebih lanjut. Visualisasi hasil dari dataset yang telah melalui tahap pemrosesan ditampilkan pada Gambar 2.

```
selected_columns = ['rad', 'temp', 'humi', 'vol-irri']
print(df[selected_columns])
```

	rad	temp	humi	vol-irri
0	0	19	90	0
1	1	19	90	0
2	0	19	91	0
3	0	19	91	0
4	0	19	91	0
...	...	...	...	...
2504	690	28	46	499
2505	675	28	48	499
2506	670	28	45	499
2507	660	28	50	529
2508	647	28	44	529

[2509 rows x 4 columns]

Gambar 2. Dataset

#### 3.2 Normalisasi

Data fitur dan target dinormalisasi menggunakan metode *Min-Max Scaling* ke dalam rentang [0, 1]. Langkah ini bertujuan untuk mempercepat proses pelatihan model serta mencegah pengaruh fitur berdasarkan skala yang lebih signifikan terhadap fitur lain dalam tahap pembelajaran. Setelah dilakukan normalisasi, seluruh nilai pada fitur input (radiasi, suhu, dan kelembapan) serta target output (volume irigasi) berhasil dikonversi dalam bentuk skala yang konsisten, berada dalam rentang 0 hingga 1. Hasil normalisasi ini memastikan bahwa kontribusi setiap fitur terhadap model berada pada tingkat yang seimbang, sehingga proses pembelajaran model *GRU* menjadi lebih stabil dan efisien.

#### 3.3 Menentukan Time Step dan Reshape Data

Pemodelan deret waktu dengan menggunakan model *GRU* memerlukan data dalam format sekuensial. Untuk memenuhi kebutuhan ini, diterapkan metode *sliding window* dengan panjang jendela (*window size*) atau *time step* sebanyak 7 hari. Artinya, model akan menggunakan data lingkungan dari tujuh hari sebelumnya untuk memprediksi volume irigasi pada hari ke-8. Selanjutnya, data diubah menjadi *array* berdimensi tiga, yaitu (*samples*,

*time steps, features*), sesuai dengan format input yang dibutuhkan oleh jaringan *GRU*. Proses ini dilakukan melalui fungsi khusus yang membentuk pasangan input-output berdasarkan urutan waktu dan kemudian dilakukan proses reshape terhadap data tersebut. Hasil dari tahapan ini adalah struktur data yang siap digunakan dalam pelatihan model *GRU*, di mana setiap sampel mewakili urutan tujuh hari data input dengan target prediksi satu hari ke depan.

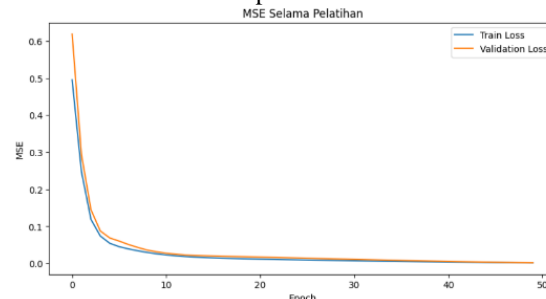
### 3.4 Pemisahan Data

Dataset awal terdiri dari 2509 baris data. Setelah dilakukan proses transformasi menggunakan pendekatan *sliding window* dengan panjang jendela sebesar 7 langkah waktu dan 3 fitur masukan, diperoleh sebanyak 2502 sampel data yang siap digunakan sebagai input model. Dataset ini dibagi menjadi dua bagian, yaitu data latih sebesar 80% (2001 sampel) dan data uji sebesar 20% (501 sampel), dengan pembagian dilakukan secara berurutan (bukan acak) untuk mempertahankan sifat temporal dari data deret waktu. Pembagian ini bertujuan untuk mengevaluasi sejauh mana model dalam menggeneralisasi pada data baru yang belum pernah dilihat sebelumnya. Dengan pendekatan ini, model *GRU* dilatih menggunakan urutan historis berdasarkan sebagian besar data, lalu diuji pada segmen data yang muncul setelahnya, sehingga hasil evaluasi mencerminkan kemampuan model dalam memprediksi kebutuhan irigasi di masa mendatang secara realistis.

### 3.5 Pelatihan Model GRU

Model *GRU* dilatih berdasarkan data latih yang telah diproses sebelumnya. Arsitektur model terbentuk dari satu lapisan *GRU* dengan 4 Unit dan fungsi aktivasi *ReLU*, diikuti oleh satu lapisan *Dense* sebagai *output* untuk menghasilkan prediksi volume irigasi. Kompilasi model dilakukan menggunakan *optimizer Adam* dengan nilai *learning rate* sebesar 0.001 dan fungsi kerugian (*loss function*) yang digunakan adalah *MSE*, karena sesuai untuk masalah regresi. Tahap pelatihan dilakukan selama 50 *epoch* dengan *batch size* sebesar 32. Selain itu, validasi dilakukan menggunakan 20% dari data uji untuk

memantau kinerja model terhadap data yang tidak dilibatkan dalam pelatihan.



Gambar 3. MSE Selama Pelatihan

### 3.6 Denormalisasi

Untuk memperoleh interpretasi hasil prediksi yang lebih bermakna dan relevan secara praktis, nilai-nilai prediksi yang telah dinormalisasi sebelumnya dikembalikan ke rentang awal melalui tahap denormalisasi. Proses ini dilakukan menggunakan fungsi *inverse\_transform* dari objek *MinMaxScaler* yang telah diterapkan pada tahap normalisasi awal. Denormalisasi ini memungkinkan hasil prediksi memiliki satuan yang sama dengan data aktual, yaitu volume irigasi dalam satuan liter (atau sesuai satuan aslinya). Langkah ini penting agar prediksi model dapat dibandingkan secara langsung dengan nilai aktual, sehingga proses evaluasi model menjadi lebih realistis dan dapat digunakan dalam aplikasi nyata di bidang pertanian berbasis *greenhouse*.

### 3.7 Evaluasi Model

Model dievaluasi menggunakan beberapa metrik performa regresi, yaitu *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, dan *R-squared (R<sup>2</sup>)*. Hasil evaluasi model *GRU* terhadap data uji disajikan pada Tabel 1 berikut.

Tabel 1. Hasil Evaluasi Model GRU

Metrik	Nilai
MAE	6.88
MSE	140.87
RMSE	11.87
R <sup>2</sup>	0.9979

Berdasarkan hasil tersebut, model *GRU* menunjukkan hasil prediksi yang sangat baik. Nilai *R<sup>2</sup>* sebesar 0.9979 menunjukkan bahwa

model mampu menjelaskan sekitar 99,79% variabilitas dari data target, yang mencerminkan kemampuan generalisasi yang sangat tinggi terhadap pola historis data. Nilai *RMSE* sebesar 11.87 mengindikasikan rata-rata deviasi antara hasil prediksi dan nilai aktual relatif rendah. Begitu pula dengan nilai *MAE* sebesar 6.88, yang menunjukkan kesalahan absolut rata-rata cukup kecil. Nilai *MSE* sebesar 140.87 mendukung temuan tersebut dengan menunjukkan bahwa kesalahan kuadrat rata-rata model juga berada pada tingkat yang rendah. Secara keseluruhan, hasil evaluasi ini mengindikasikan model *GRU* sangat efektif dalam memprediksi kebutuhan volume irigasi pada sistem pertanian *greenhouse*.

### 3.8 Perbandingan Nilai Aktual Vs Prediksi

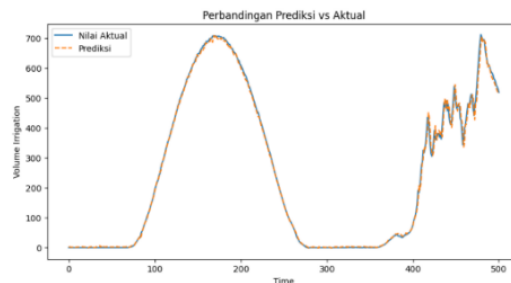
Untuk mengevaluasi akurasi model secara menyeluruh, dilakukan analisis terhadap hasil prediksi model *GRU* dengan membandingkannya terhadap nilai aktual volume irigasi. Evaluasi dilakukan baik secara numerik maupun visual untuk mendapatkan gambaran yang komprehensif mengenai performa model. Secara numerik, Tabel 2 menyajikan sebagian hasil prediksi model terhadap 100 data uji terakhir yang telah melalui proses denormalisasi. Perbandingan ini penting untuk mengamati seberapa besar deviasi antara prediksi dan data sebenarnya, terutama pada nilai-nilai ekstrem atau titik perubahan signifikan.

Tabel 2. Perbandingan Nilai Aktual dan Prediksi Model *GRU*

Indeks	Nilai Aktual	Nilai Prediksi
0	0.00	1.44
1	0.00	3.04
2	0.00	1.62
3	0.00	1.45
4	0.00	1.46
...	...	...
95	140.11	134.98
96	148.12	149.22
97	156.93	158.39
98	168.94	166.49
99	180.15	180.66

Sementara itu, secara visual, Gambar 3 menampilkan grafik perbandingan antara nilai aktual dan prediksi dalam bentuk deret waktu. Terlihat bahwa kurva prediksi (garis oranye

putus-putus) mengikuti pola kurva aktual (garis biru) dengan baik. Model *GRU* mampu menangkap pola umum, tren, dan fluktuasi yang terjadi pada volume irigasi dari waktu ke waktu.



Gambar 4. Perbandingan prediksi vs aktual

Gabungan antara analisis numerik dan visualisasi ini mendukung hasil evaluasi metrik sebelumnya, bahwa model *GRU* tidak hanya memberikan performa statistik yang baik, tetapi juga mampu menghasilkan prediksi yang representatif terhadap kondisi nyata di lapangan.

## 4. KESIMPULAN

Dari hasil penelitian, disimpulkan bahwa prediksi kebutuhan air untuk penyiraman tanaman paprika dalam sistem *greenhouse* dapat dilakukan dengan menggunakan metode *GRU*. Model ini dirancang untuk memproses data deret waktu berdasarkan parameter lingkungan seperti suhu, kelembapan, dan radiasi, sehingga mampu menangkap pola historis yang memengaruhi kebutuhan irigasi tanaman. Akurasi model *GRU* dalam memprediksi kebutuhan air menunjukkan hasil yang sangat baik, dengan nilai  $R^2$  sebesar 0.9979, *MAE* sebesar 6.88, *MSE* sebesar 140.87, dan *RMSE* sebesar 11.87. Nilai-nilai ini mengindikasikan bahwa model *GRU* mampu melakukan prediksi secara akurat dengan deviasi yang rendah terhadap data aktual. Dengan demikian, model *GRU* yang dikembangkan dalam penelitian ini telah terbukti efektif dalam menjawab rumusan masalah, dan dapat direkomendasikan untuk diterapkan dalam sistem irigasi otomatis berbasis teknologi cerdas guna mendukung efisiensi penggunaan air di pertanian *greenhouse*.

---

**DAFTAR PUSTAKA**

- [1] *The State of Food and Agriculture 2022*. FAO, 2022. doi: 10.4060/cb9479en.
- [2] M. Hejazi *et al.*, “Impacts of water scarcity on agricultural production and electricity generation in the Middle East and North Africa,” *Front Environ Sci*, vol. 11, 2023, doi: 10.3389/fenvs.2023.1082930.
- [3] B. Basso and J. Antle, “Digital agriculture to design sustainable agricultural systems,” Apr. 01, 2020, *Nature Research*. doi: 10.1038/s41893-020-0510-0.
- [4] C. Maraveas, C. S. Karavas, D. Loukatos, T. Bartzanas, K. G. Arvanitis, and E. Symeonaki, “Agricultural Greenhouses: Resource Management Technologies and Perspectives for Zero Greenhouse Gas Emissions,” 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/agriculture13071464.
- [5] F. Fuentes-Peñaillillo, K. Gutter, R. Vega, and G. C. Silva, “New Generation Sustainable Technologies for Soilless Vegetable Production,” Jan. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/horticulturae10010049.
- [6] V. Mamatha and J. C. Kavitha, “Machine learning based crop growth management in greenhouse environment using hydroponics farming techniques,” *Measurement: Sensors*, vol. 25, Feb. 2023, doi: 10.1016/j.measen.2023.100665.
- [7] A. Syukur, R. Zega, A. Kurnia Hidayat, N. T. Jannah, and F. Kartiasih, “SELECTING THE BEST MODEL FOR FORECASTING INDONESIA’S OIL AND GAS IMPORT VALUE USING ARIMAX AND ARIMAX-LSTM,” *Dynamic Management Journal*, vol. 8, no. 4, 2024, doi: 10.31000/dmj.v8i4.
- [8] G. T. Michael, M. Turnip, E. Muniarti, E. Sitompul, and A. Turnip, “Development of an Irrigation System for Predicting Watering Time with ANFIS Method for Chili Plants,” in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, 2022. doi: 10.1088/1755-1315/1083/1/012081.
- [9] D. Maria Manuel Vianny, A. John, S. Kumar Mohan, A. Sarlan, Adimoolam, and A. Ahmadian, “Water optimization technique for precision irrigation system using IoT and machine learning,” *Sustainable Energy Technologies and Assessments*, vol. 52, Aug. 2022, doi: 10.1016/j.seta.2022.102307.
- [10] D. Kurniasari, S. Oskavina, W. Wamiliana, and W. Warsono, “Forecasting The Value of Indonesian Oil-Non-Oil and Gas Imported Using The Gated Recurrent Unit (GRU),” *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 6, no. 1, p. 71, Jul. 2023, doi: 10.24014/ijaidm.v6i1.20651.
- [11] F. He, Q. Wan, Y. Wang, J. Wu, X. Zhang, and Y. Feng, “Daily Runoff Prediction with a Seasonal Decomposition-Based Deep GRU Method,” *Water (Switzerland)*, vol. 16, no. 4, Feb. 2024, doi: 10.3390/w16040618.
- [12] J. Nie, N. Wang, J. Li, Y. Wang, and K. Wang, “Prediction of Liquid Magnetization Series Data in Agriculture Based on Enhanced CGAN,” *Front Plant Sci*, vol. 13, Jun. 2022, doi: 10.3389/fpls.2022.929140.
- [13] S. Amassmir, S. Tkatek, O. Abdoun, and J. Abouchabaka, “An intelligent irrigation system based on internet of things to minimize water loss,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 1, pp. 504–510, Jan. 2022, doi: 10.11591/ijeecs.v25.i1.pp504-510.
- [14] I. P. G. A. Sudiatmika, I. M. A. W. Putra, and W. W. Artana, “The Implementation of Gated Recurrent Unit (GRU) for Gold Price Prediction Using Yahoo Finance Data: A Case Study and Analysis,” *Brilliance: Research of Artificial Intelligence*, vol. 4, no. 1, pp. 176–184, Jun. 2024, doi: 10.47709/brilliance.v4i1.3865.

