

**PENERAPAN *GATED RECURRENT UNIT* (GRU) UNTUK
PREDIKSI JUMLAH KASUS DEMAM BERDARAH DENGUE (DBD)
(STUDI KASUS: KOTA MALANG)**

SKRIPSI

**OLEH
SILVI NUR AFIFA DARMAYANTI
NIM. 220601110099**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026**

**PENERAPAN *GATED RECURRENT UNIT* (GRU) UNTUK
PREDIKSI JUMLAH KASUS DEMAM BERDARAH DENGUE (DBD)
(STUDI KASUS: KOTA MALANG)**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh
Silvi Nur Afifa Darmayanti
NIM. 220601110099**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026**

**PENERAPAN *GATED RECURRENT UNIT* (GRU) UNTUK
PREDIKSI JUMLAH KASUS DEMAM BERDARAH DENGUE (DBD)
(STUDI KASUS: KOTA MALANG)**

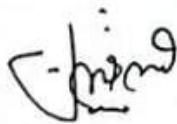
SKRIPSI

Oleh
Silvi Nur Afifa Darmayanti
NIM. 220601110099

Telah Disetujui untuk Diuji

Malang, 17 April 2026

Dosen Pembimbing I



Ria Dhea Layla Nur Karisma, M.Si
NIPPPK. 19900709 202321 2 037

Dosen Pembimbing II



Mohammad Nafie Jauhari, M.Si
NIPPPK. 19870218 202321 1 018

Mengetahui,
Ketua Program Studi Matematika



Dr. Faehur Rozi, M.Si
NIP. 19800527 200801 1 012

**PENERAPAN *GATED RECURRENT UNIT* (GRU) UNTUK
PREDIKSI JUMLAH KASUS DEMAM BERDARAH DENGUE (DBD)
(STUDI KASUS: KOTA MALANG)**

SKRIPSI

Oleh
Silvi Nur Afifa Darmayanti
NIM. 220601110099

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

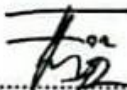
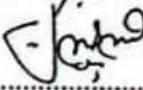
Tanggal 26 Mei 2026

Ketua Penguji : Dr. Mohammad Jamhuri, M.Si

Anggota Penguji 1 : Dr. Fachrur Rozi, M.Si

Anggota Penguji 2 : Ria Dhea Layla Nur Karisma, M.Si

Anggota Penguji 3 : Mohammad Nafie Jauhari, M.Si


.....

.....

.....

.....

Mengetahui,
Ketua Program Studi Matematika



Dr. Fachrur Rozi, M.Si
NIP. 19800527 200801 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Silvi Nur Afifa Darmayanti
NIM : 220601110099
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Penerapan *Gated Recurrent Unit* (GRU) untuk Prediksi Jumlah Kasus Demam Berdarah Dengue (DBD) (Studi Kasus: Kota Malang)

Menyatakan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri dan bukan hasil pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai hasil tulisan atau pemikiran saya sendiri, kecuali dengan mencantumkan sumber kutipan dalam daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini merupakan hasil plagiasi, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 Mei 2026

Yang membuat pernyataan,



Silvi Nur Afifa Darmayanti

NIM. 220601110099

MOTO

“Seseorang yang Bersabar Tidak Akan Pernah Kehilangan Kesuksesan Meskipun
Membutuhkan Waktu yang Lama untuk Mencapainya”

(Ali Bin Abi Thalib)

“Dan Bersabarlah Kamu. Sesungguhnya Janji Allah adalah Benar”

(Q.S Ar-Rum : 60)

PERSEMBAHAN

Alhamdulillahirabbil'alamin

Puji dan syukur tiada hentinya terhanturkan kepada Allah SWT

Kupersembahkan karya ini kepada:

Orang Tua

Sosok yang tak pernah lelah mengetuk pintu langit, Ayahanda Darmaji dan Ibunda Dami Ariyanti. Terima kasih atas setiap butir doa istiqomah yang menjadi cahaya di saat penulis merasa gelap, dan atas nasihat-nasihat yang menjadi kompas di kala penulis kehilangan arah. Perjuangan ini adalah tentang kalian, tentang cinta kalian yang tak ternilai harganya.

Saudara

Kepada saudaraku tercinta, Siti Nur Arifah Dharma Putri, Bagus Supriyanto, dan M. Hisyam Faza Syahputra. Terima kasih telah senantiasa menjadi alasan bagi penulis untuk terus berusaha memberikan yang terbaik.

Almamater Kebanggaan

Universitas Islam Negeri Maulana Malik Ibrahim Malang.

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh.

Alhamdulillah robbil ‘alamin, puji syukur ke hadirat Allah SWT atas segala limpahan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Penerapan *Gated Recurrent Unit* (GRU) untuk Prediksi Jumlah Kasus Demam Berdarah Dengue (DBD) (Studi Kasus: Kota Malang)”. Penyusunan skripsi ini merupakan salah satu syarat untuk menyelesaikan studi pada Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulis menyadari sepenuhnya bahwa penyusunan skripsi ini tidak akan terselesaikan tanpa adanya doa, dukungan, arahan, serta bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrur Rozi, M.Si, selaku Ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Ria Dhea Layla Nur Karisma, M.Si, selaku Dosen Pembimbing I yang telah meluangkan waktunya untuk memberikan bimbingan dan arahan terbaik agar penulis dapat menyelesaikan skripsi dengan baik.
5. Mohammad Nafie Jauhari, M.Si, selaku Dosen Pembimbing II yang telah memberikan nasehat dan bimbingan terbaik kepada penulis.
6. Seluruh Dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan ilmu dan bantuan selama perkuliahan.
7. Kedua orang tua tercinta, Ayahanda Darmaji dan Ibunda Dami Ariyanti, sebagai pilar utama kekuatan penulis. Terima kasih atas doa yang tidak pernah putus, dukungan finansial, kasih sayang yang tulus, serta motivasi yang selalu membangkitkan semangat penulis di saat terpuruk. Skripsi ini penulis persembahkan untuk kalian.

Secara khusus, penulis juga ingin menyampaikan rasa terima kasih kepada pihak-pihak yang telah memberikan warna dan dukungan emosional yang luar biasa selama proses ini:

8. Kepada teman terdekat saya, Keisya, Aulia, Intan, dan Nisa'. Terima kasih atas tawa yang meringankan beban, gangguan-gangguan kecil yang justru membuat stres menghilang, dan karena tetap ada di samping penulis meski penulis sering kali sulit dihubungi karena tenggelam dalam revisi. Pertemanan ini adalah salah satu pencapaian terbaik selama masa kuliah.
9. Kepada Kak Nanda atau dikenal dengan Qim, seseorang yang sabar menghadapi dinamika emosional penulis selama penyusunan skripsi ini. Terima kasih telah menjadi pendengar setia saat penulis merasa *down*, memberikan semangat di saat rasa malas melanda, dan selalu meyakinkan penulis bahwa "hari kelulusan itu pasti datang". Kehadiranmu, pengertianmu, serta dukungan moral yang tiada henti adalah energi tambahan yang sangat berarti bagi penulis.
10. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu baik secara langsung maupun tidak langsung dalam penyelesaian tugas akhir ini.

Semoga Allah SWT membalas segala kebaikan dan ketulusan yang telah diberikan kepada penulis. Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat berbagai kekurangan, sehingga dengan lapang dada penulis menerima setiap kritik dan saran yang bersifat membangun dari berbagai pihak. Penulis juga berharap, karya ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi para pembaca, serta dapat dijadikan sebagai referensi dan menambah wawasan ilmu pengetahuan.

Wassalamu 'alaikum warahmatullahi wabarakatuh.

Malang, 26 Mei 2026

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
DAFTAR SIMBOL	xv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	6
1.3 Tujuan Penelitian	6
1.4 Manfaat Penelitian.....	7
1.5 Batasan Masalah	8
1.6 Definisi Istilah	8
BAB II KAJIAN TEORI	10
2.1 Teori Pendukung.....	10
2.1.1 <i>Preprocessing</i> Data	10
2.1.2 <i>Gated Recurrent Unit</i> (GRU).....	11
2.1.2.1 Mekanisme GRU	12
2.1.2.2 <i>Hyperparameter</i> GRU	15
2.1.3 Kriteria Evaluasi	19
2.1.3.1 <i>Mean Absolute Percentage Error</i> (MAPE)	19
2.1.4 Demam Berdarah Dengue (DBD).....	20
2.1.5 Curah Hujan.....	21
2.1.6 Suhu Udara	22
2.1.7 Kelembapan Udara	23
2.2 Kajian Integrasi Topik dengan Al-Quran/Hadits	23
2.3 Kajian Topik dengan Teori Pendukung	26
BAB III METODE PENELITIAN	29
3.1 Jenis Penelitian	29
3.2 Data dan Sumber Data	29
3.3 Teknik Pengumpulan Data.....	29
3.4 Variabel Penelitian	30
3.5 Teknik Analisis Data.....	32
3.6 Diagram Alir Penelitian	35
BAB IV HASIL DAN PEMBAHASAN.....	36

4.1 Statistika Deskriptif	36
4.2 <i>Preprocessing</i> Data	40
4.3 Inisialisasi <i>Hyperparameter</i>	44
4.4 Proses <i>Training</i> GRU	45
4.5 Perhitungan Matematis Model GRU	47
4.6 Hasil Prediksi Jumlah DBD	75
4.7 Evaluasi Model	77
4.8 Integrasi Hasil Penelitian dengan Kajian Agama	78
BAB V PENUTUP	80
5.1 Kesimpulan	80
5.2 Saran	81
DAFTAR PUSTAKA	82
LAMPIRAN	86
DAFTAR RIWAYAT HIDUP	95

DAFTAR TABEL

Tabel 2.1	Kriteria Nilai MAPE	19
Tabel 3.1	Variabel dan Operasionalisasi Data	30
Tabel 3.2	Kombinasi <i>Hyperparameter</i>	33
Tabel 4.1	Statistika Deskriptif Data	36
Tabel 4.2	Statistika Deskriptif Hasil Normalisasi Data <i>Training</i>	41
Tabel 4.3	Statistika Deskriptif Hasil Normalisasi Data <i>Testing</i>	42
Tabel 4.4	Uji Coba Penentuan <i>Timestep</i>	43
Tabel 4.5	<i>Reshaping</i> Data GRU	44
Tabel 4.6	Inisialisasi <i>Hyperparameter</i> GRU	45
Tabel 4.7	Nilai Iterasi <i>Epoch</i> dan <i>Loss</i>	45
Tabel 4.8	Parameter Arsitektur Model GRU (<i>Stacked</i> GRU)	75
Tabel 4.9	Hasil Prediksi dan Data Aktual DBD	75
Tabel 4.10	Perbandingan Nilai MAPE Model Prediksi DBD	77

DAFTAR GAMBAR

Gambar 2.1 Arsitektur GRU (Tu dkk., 2024)	12
Gambar 3.1 Diagram Alir Penelitian.....	35
Gambar 4.1 Plot Data Kasus DBD.....	37
Gambar 4.2 Plot Data Curah Hujan	38
Gambar 4.3 Plot Data Suhu Udara	39
Gambar 4.4 Plot Data Kelembapan Udara	40
Gambar 4.5 Plot Iterasi <i>Epoch</i> dan <i>Loss</i>	46
Gambar 4.6 Plot Hasil Prediksi dan Data Aktual	76

DAFTAR LAMPIRAN

Lampiran 1. Data Jumlah Kasus DBD dan Faktor Iklim	86
Lampiran 2. Arsitektur GRU untuk Prediksi Kasus DBD	90
Lampiran 3. <i>Syntax</i> Prediksi Kasus DBD Menggunakan Metode GRU	91

DAFTAR SIMBOL

x'	: Hasil data setelah dinormalisasi
x	: Data yang dinormalisasi
x_{min}	: Nilai minimum dari keseluruhan data x
x_{max}	: Nilai maksimum dari keseluruhan data x
x_t	: Vektor input pada waktu (t)
h_t	: Vektor keadaan tersembunyi (<i>hidden state</i>) pada waktu (t)
W_z	: Matriks bobot input untuk <i>update gate</i>
W_r	: Matriks bobot input untuk <i>reset gate</i>
W_h	: Matriks bobot input untuk <i>candidate hidden state</i>
W_o	: Matriks bobot yang menghubungkan <i>hidden state</i> ke <i>layer output</i>
U_z	: Matriks bobot <i>recurrent</i> untuk <i>update gate</i> pada <i>hidden state</i> sebelumnya
U_r	: Matriks bobot <i>recurrent</i> untuk <i>reset gate</i> pada <i>hidden state</i> sebelumnya
U_h	: Matriks bobot <i>recurrent</i> untuk <i>candidate hidden state</i> pada <i>hidden state</i> sebelumnya
b_z	: Vektor bias untuk <i>update gate</i>
b_r	: Vektor bias untuk <i>reset gate</i>
b_h	: Vektor bias untuk <i>candidate hidden state</i>
σ	: Fungsi aktivasi sigmoid/fungsi aktivasi (tanh atau sigmoid)
$\tanh$	: Fungsi tangens hiperbolik
y_t	: Nilai aktual
$\hat{y}_t$	: Nilai hasil prediksi pada waktu ke-t
r_t	: Vektor <i>reset gate</i>
z_t	: Vektor <i>update gate</i>
$\odot$	: Operasi perkalian elemen demi elemen
h	: Output setelah dropout diterapkan
η	: Penentuan seberapa besar pembaruan bobot pada tiap iterasi
g_t	: Turunan dari <i>loss</i> terhadap bobot
m_t	: Matriks estimasi momen pertama (<i>mean</i>)
v_t	: Matriks estimasi momen kedua (<i>Uncentered Variance</i>)
θ	: Parameter yang akan dioptimalkan
β_1, β_2	: Koefisien untuk eksponensial <i>moving average</i>
ε	: Nilai kecil untuk menghindari pembagian nol

ABSTRAK

Darmayanti, Silvi Nur Afifa. 2026. **Penerapan Gated Recurrent Unit (GRU) untuk Prediksi Jumlah Kasus Demam berdarah Dengue (DBD) (Studi Kasus: Kota Malang)**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing: (I) Ria Dhea Layla Nur Karisma, M.Si. (II) Mohammad Nafie Jauhari, M.Si.

Kata Kunci: Demam Berdarah Dengue (DBD), *Gated Recurrent Unit* (GRU), Iklim, MAPE, Prediksi, SARIMAX.

Penyakit Demam Berdarah Dengue (DBD) di Kota Malang memiliki pola fluktuasi yang dipengaruhi oleh berbagai faktor, termasuk unsur iklim. Penelitian ini bertujuan memprediksi jumlah kasus DBD dengan membandingkan model *Gated Recurrent Unit* (GRU) dan SARIMAX menggunakan variabel *lag* kasus DBD, curah hujan, suhu udara, dan kelembapan. Data diproses melalui tahap prapengolahan berupa normalisasi dan transformasi menggunakan metode *sliding window* pada model GRU. Hasil penelitian menunjukkan bahwa model GRU mampu mengenali pola hubungan antara data historis dan data aktual dalam memprediksi kasus DBD. Namun, berdasarkan hasil evaluasi menggunakan *Mean Absolute Percentage Error* (MAPE), model SARIMAX(1,1,1)(1,1,0)₁₂ menghasilkan performa yang lebih baik dengan nilai MAPE sebesar 21.58%, dibandingkan model GRU yang memperoleh nilai MAPE sebesar 29.29%. Hasil tersebut menunjukkan bahwa pola musiman pada data kasus DBD lebih dominan sehingga lebih efektif dimodelkan menggunakan pendekatan SARIMAX. Dengan demikian, model SARIMAX dapat digunakan sebagai instrumen prediksi untuk mendukung upaya pencegahan dan pengendalian DBD di Kota Malang.

ABSTRACT

Darmayanti, Silvi Nur Afifa. 2026. **Application of Gated Recurrent Unit (GRU) to Predict the Number of Dengue Fever (DHF) Cases (Case Study: Malang City).** Undergraduate Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Advisors: (I) Ria Dhea Layla Nur Karisma, M.Si. (II) Mohammad Nafie Jauhari, M.Si.

Keywords: Dengue Fever (DHF), Gated Recurrent Unit (GRU), Climate, MAPE, Prediction, SARIMAX.

Dengue Hemorrhagic Fever (DHF) in Malang City exhibits fluctuating patterns influenced by various factors, including climate variables. This study aims to predict the number of DHF cases by comparing the Gated Recurrent Unit (GRU) and SARIMAX models using lagged DHF cases, rainfall, air temperature, and humidity as predictors. Data preprocessing involved normalization and transformation using the sliding window method for the GRU model. The results indicate that the GRU model is capable of identifying relationships between historical and actual data in predicting DHF cases. However, based on the evaluation using *Mean Absolute Percentage Error* (MAPE), the SARIMAX(1,1,1)(1,1,0)₁₂ model achieved better performance with a MAPE value of 21.58%, compared to the GRU model with a MAPE value of 29.29%. These findings suggest that seasonal patterns in DHF case data are more dominant and can be modeled more effectively using the SARIMAX approach. Therefore, the SARIMAX model can be utilized as a predictive tool to support dengue prevention and control efforts in Malang City.

مستخلص البحث

دارمايانتى، سيلفي نور عفيفة . ٢٠٢٦ . تطبيق وحدة التكرار الموجه (*GRU*) للتنبؤ بعدد حالات حمى الضنك النزفية (*DBD*) (دراسة حالة: مدينة مالانغ). البحث العلمي ، قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانغ.
المشرف: (1) ريا دينا ليلي نور كرسما، الماجستير في العلوم ، (2) محمد نافع جواهري ، الماجستير في العلوم .

الكلمات الرئيسية: حمى الضنك النزفية (*DBD*)، وحدة التكرار الموجه (*GRU*)، المناخ، متوسط نسبة الخطأ المطلق (*MAPE*)، التنبؤ.

في مدينة مالانغ نغطاً متقلّباً يتأثر بعوامل متعددة، بما في ذلك العناصر (*DBD*) يُظهر مرض حمى الضنك النزفية *Gated Recurrent Unit (GRU)* المناخية تهدف هذه الدراسة إلى التنبؤ بعدد حالات حمى الضنك النزفية من خلال مقارنة نموذج باستخدام متغيرات تأخر حالات حمى الضنك، ومعدل هطول الأمطار *SARIMAX* ونموذج *Unit (GRU)* ودرجة حرارة الهواء، والرطوبة. تمت معالجة البيانات من خلال مرحلة المعالجة المسبقة التي شملت التطبيع والتحويل باستخدام قادر على التعرف على *GRU* وأظهرت نتائج الدراسة أن نموذج *GRU* في نموذج *Sliding Window* طريقة أنماط العلاقة بين البيانات التاريخية والبيانات الفعلية في التنبؤ بحالات حمى الضنك النزفية. ومع ذلك، واستناداً إلى نتائج $12(1,1,1)$ $12(1,1,1)$ *SARIMAX* فقد أظهر نموذج (*MAPE*) التقييم باستخدام متوسط نسبة الخطأ المطلق $29,29$ *MAPE* الذي حقق قيمة *GRU* بلغت $21,58$ ٪ مقارنةً بنموذج *MAPE* أداءً أفضل بقيمة $29,29$ ٪. وتشير هذه النتائج إلى أن النمط الموسمي في بيانات حالات حمى الضنك النزفية أكثر هيمنة، مما يجعله أكثر فاعلية عند كأداة تنبؤية لدعم جهود *SARIMAX* وبالتالي، يمكن استخدام نموذج *SARIMAX* تمذجته باستخدام منهج الوقاية من حمى الضنك النزفية والسيطرة عليها في مدينة مالانغ

BAB I

PENDAHULUAN

1.1 Latar Belakang

Metode *Gated Recurrent Unit* (GRU) merupakan salah satu arsitektur dalam jaringan saraf tiruan yang dirancang untuk memproses data berurutan, sebagai pengembangan dari *Recurrent Neural Network* (RNN) untuk mengatasi masalah *vanishing gradient* yang sering terjadi pada jaringan saraf tradisional. Menurut Goodfellow dkk (2016), *vanishing gradient* terjadi ketika gradien yang dihasilkan oleh fungsi aktivasi menjadi sangat kecil atau menghilang saat menyebar ke lapisan-lapisan yang lebih dalam, menghambat pembelajaran hubungan jangka panjang. GRU menggunakan fungsi aktivasi sigmoid pada *update gate* dan *reset gate* untuk mengatur aliran informasi, serta fungsi tanh untuk menghitung *candidate hidden state*, memungkinkan penyimpanan informasi yang relevan secara selektif (Cho dkk., 2014).

RNN sendiri rentan terhadap masalah *vanishing gradient*, di mana gradien menjadi sangat kecil selama *backpropagation* yaitu algoritma pelatihan yang digunakan dalam jaringan saraf tiruan untuk mengoptimalkan parameter jaringan seperti bobot dan bias, dengan meminimalkan fungsi kerugian (*loss function*), sehingga sulit untuk mempelajari hubungan jangka panjang. Sedangkan GRU dapat mengatasi masalah ini dengan mekanisme gerbang yaitu *update gate* dan *reset gate* yang lebih mampu menangani hubungan jangka panjang dan lebih stabil karena mekanisme gerbangnya mencegah *vanishing gradient* (Noh, 2021).

GRU merupakan varian dari *Long Short-Term Memory* (LSTM) yang dirancang dengan arsitektur lebih sederhana namun tetap mampu menangkap dependensi jangka panjang pada data deret waktu. Dengan hanya menggunakan dua gerbang utama, yaitu *update gate* dan *reset gate*, tanpa adanya *cell state* terpisah seperti pada LSTM, GRU mampu mempercepat proses pelatihan dan mengurangi kompleksitas model. Keunggulan ini menjadikan GRU lebih efisien secara komputasi dan lebih sesuai untuk dataset dengan ukuran terbatas, sementara tetap mempertahankan performa yang kompetitif dalam mempelajari pola temporal yang kompleks (Meriani & Rahmatulloh, 2024).

Menurut Cho dkk. (2014), GRU efektif dalam memprediksi data deret waktu karena kemampuannya menyimpan informasi jangka panjang dan mengabaikan informasi yang tidak relevan. Dengan demikian, GRU menjadi solusi adaptif untuk menangkap hubungan pada skala waktu yang berbeda-beda, dan telah terbukti efektif dalam berbagai aplikasi seperti peramalan deret waktu harga emas dan kripto di Indonesia (Melina dkk., 2025).

Rochman dkk. (2022) dalam penelitiannya menggunakan data pasien dari Puskesmas Modopuro, Kecamatan Mojosari, Kabupaten Mojokerto. Proses prediksi dilakukan dengan cara melakukan proses perhitungan data secara manual, lalu mencari korelasi data sesuai dengan metode LSTM, yaitu mengidentifikasi koefisien autokorelasi pada dua sampai tiga *time-lag* yang berbeda. Data yang diamati merupakan data harian sebanyak 345 data. Dari hasil perhitungan tersebut didapatkan nilai RMAPE sebesar 3.184 sedangkan GRU menghasilkan RMAPE sebesar 1.747. Dengan demikian GRU dapat dikatakan lebih baik dalam memprediksi jumlah kunjungan pasien penyakit diabetes dibandingkan LSTM.

Penelitian ini dilakukan untuk menjawab permasalahan tersebut dengan menerapkan model GRU sebagai metode prediksi jumlah kasus Demam Berdarah Dengue (DBD). Dengan memanfaatkan data historis, GRU diharapkan mampu mengenali pola peningkatan kasus secara lebih tepat. DBD sendiri merupakan penyakit endemik di Indonesia yang muncul setiap tahun dan cenderung menunjukkan tren kenaikan kasus. Pada minggu ke-17 tahun 2024, tercatat sebanyak 88.593 kasus dengan 621 kematian. Meskipun sempat mengalami penurunan sekitar 35% pada tahun 2023 hingga awal 2024, jumlah kasus kembali meningkat pada minggu ke-22 tahun 2024 menjadi 119.709, lebih tinggi dari tahun sebelumnya yang mencatat 114.720 kasus (Kemenkes RI, 2024). Di Kota Malang, hingga Mei 2025 telah tercatat 389 kasus DBD, beberapa di antaranya menyebabkan kematian. Sementara itu, sepanjang tahun 2024 jumlah kasus DBD mencapai 777 dengan 4 kematian (Dinkes Kota Malang, 2025).

Penelitian Rakhmatsani, dkk. (2024) menunjukkan bahwa faktor cuaca seperti curah hujan, suhu, dan kelembapan, serta kepadatan penduduk, memiliki pengaruh yang signifikan terhadap pola penyebaran DBD di kawasan perkotaan. Adapun firman Allah SWT pada QS. Yunus [10]: 101 (Kemenag, 2024), sebagai berikut.

قُلْ اَنْظُرُوْا مَاذَا فِي السَّمٰوٰتِ وَالْاَرْضِ وَمَا تُعْنٰی الْاٰیٰتِ وَالتَّنٰذِرِ عَنْ قَوْمٍ لَا يُؤْمِنُوْنَ ﴿١٠١﴾

“Katakanlah: Perhatikanlah apa yang ada di langit dan di bumi! Tetapi tanda-tanda (kekuasaan Allah) dan semua peringatan itu tidak berguna bagi orang-orang yang tidak beriman.” (QS. Yunus [10]: 101)

Ayat tersebut mengandung pesan bahwa manusia diperintahkan untuk memperhatikan, meneliti, dan merenungkan segala ciptaan Allah yang ada di langit dan di bumi. Islam menekankan pentingnya penggunaan akal dan kemampuan berpikir kritis dalam memahami fenomena alam sebagai sarana mengenal

kebesaran Allah SWT. Dalam konteks penelitian ilmiah, perintah ini dapat dimaknai sebagai dorongan agar manusia terus melakukan observasi, pengkajian, dan analisis terhadap tanda-tanda alam secara sistematis. Dengan demikian, hasil pengamatan yang diolah menjadi pengetahuan dapat dimanfaatkan untuk kemaslahatan hidup, sejalan dengan semangat ilmiah yang dianjurkan dalam Islam.

Apabila dikaitkan dengan penelitian prediksi kasus DBD, kandungan ayat ini menegaskan bahwa memahami pola penyebaran penyakit juga merupakan bentuk upaya membaca tanda-tanda alam yang Allah ciptakan. Fenomena meningkatnya kasus DBD tidak terjadi tanpa sebab, melainkan dipengaruhi oleh faktor-faktor lingkungan seperti curah hujan, suhu, kelembapan, dan kepadatan penduduk. Melalui pendekatan ilmiah dengan bantuan algoritma GRU, penelitian ini berusaha menafsirkan pola data temporal yang kompleks agar dapat memprediksi potensi peningkatan kasus DBD secara lebih akurat dan tepat waktu. Dengan demikian, penelitian ini tidak hanya berorientasi pada hasil teknis semata, tetapi juga mencerminkan integrasi antara sains dan nilai-nilai keislaman sebagaimana terkandung dalam QS Yunus ayat 101, sebagai bentuk pengabdian manusia kepada Allah SWT melalui pengembangan ilmu pengetahuan bermanfaat bagi masyarakat.

Penelitian oleh Othman dkk. (2022) menunjukkan bahwa pendekatan hybrid berbasis SARIMA, ARIMA, dan LSTM efektif dalam meramalkan jumlah kasus Demam Berdarah Dengue (DBD) di Surabaya pada tahun 2014 hingga 2016. Model SARIMA $(2,1,1)(1,0,0)_{12}$ menghasilkan akurasi terbaik dengan RMAPE terendah dan MAE sebesar 18.97, meningkatkan performa prediksi hingga 25% dibandingkan LSTM konvensional, terutama dengan memanfaatkan data iklim sebagai variabel tambahan untuk *forecasting* hingga 12 bulan ke depan.

Data kasus DBD yang bersifat kontinu dan dinamis membutuhkan metode prediksi kompleks yang mampu menangani ketidakpastian, seperti algoritma Jaringan Syaraf Tiruan (JST) jenis *backpropagation*. Data bulanan jumlah pasien DBD dalam satu tahun digunakan sebagai input, dengan prediksi jumlah pasien untuk bulan berikutnya sebagai output. Arsitektur *backpropagation* terdiri dari 12 neuron pada lapisan input, satu lapisan tersembunyi dengan jumlah neuron yang dapat disesuaikan, dan satu neuron pada lapisan output. Hasil pengujian menunjukkan nilai MAPE terkecil sebesar 4.48 pada alfa 0.7 dengan 50 neuron tersembunyi, mencapai akurasi prediksi sebesar 95.52% (Riswanto dkk., 2014).

Penulis tertarik meneliti topik Demam Berdarah Dengue (DBD) karena menyadari dampak signifikan penyakit ini terhadap kesehatan masyarakat dan ekonomi di Indonesia. Secara pribadi, penulis melihat pentingnya inovasi dalam metode prediksi sebagai bagian dari upaya pencegahan dan pengendalian wabah DBD, mengingat DBD merupakan penyakit endemis yang dipengaruhi oleh faktor iklim dan lingkungan yang kompleks. Dari sisi akademis, penelitian ini diharapkan dapat memberikan kontribusi penting dalam pengembangan *machine learning*, khususnya dengan menerapkan model *Gated Recurrent Unit* (GRU) yang terbukti efektif dalam menangkap pola data waktu dan memberikan prediksi yang lebih akurat. Penelitian serupa menegaskan bahwa penggunaan model *machine learning*, termasuk *Gated Recurrent Unit* (GRU), dapat meningkatkan akurasi prediksi wabah penyakit menular seperti DBD, sehingga mendukung upaya pengendalian penyakit secara efektif (Nasir dkk., 2024).

Kontribusi penelitian ini dalam bidang Matematika terletak pada penerapan analisis deret waktu melalui arsitektur *Gated Recurrent Unit* (GRU) untuk

memodelkan serta memprediksi jumlah kasus Demam Berdarah Dengue (DBD). Penelitian ini mengintegrasikan konsep statistika, optimasi numerik, jaringan saraf tiruan, dan *machine learning* dalam membangun model prediksi berbasis data historis dan faktor iklim seperti curah hujan, suhu, dan kelembapan. Selain itu, penelitian ini juga memberikan kontribusi pada pengembangan kajian matematika terapan khususnya dalam pemodelan prediktif kesehatan masyarakat dengan memanfaatkan teknik preprocessing data, fungsi aktivasi, optimasi Adam, serta evaluasi akurasi menggunakan MAPE. Hasil penelitian menunjukkan bahwa metode GRU mampu menangkap pola temporal secara efektif sehingga dapat menjadi referensi akademik dalam pengembangan model prediksi berbasis kecerdasan buatan pada bidang statistika dan matematika komputasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang, didapatkan rumusan masalah yaitu:

1. Bagaimana penerapan *Gated Recurrent Unit* untuk prediksi jumlah kasus Demam Berdarah Dengue (DBD)?
2. Bagaimana kinerja model *Gated Recurrent Unit* untuk memprediksi jumlah kasus Demam Berdarah Dengue (DBD) yang diukur menggunakan MAPE?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, peneliti memiliki tujuan sebagai berikut:

1. Mengimplementasikan arsitektur *Gated Recurrent Unit* (GRU) dalam membangun model prediksi jumlah kasus Demam Berdarah Dengue (DBD).

2. Mengevaluasi performa model *Gated Recurrent Unit* (GRU) dalam memprediksi jumlah kasus Demam Berdarah Dengue (DBD) dengan menggunakan metrik *Mean Absolute Percentage Error* (MAPE).

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat, sebagai berikut:

1. Bagi Program Studi Matematika dan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang
 - a. Memberikan bahan literatur dan referensi akademik untuk dikembangkan oleh mahasiswa atau peneliti lain terutama dalam bidang statistika.
 - b. Membantu menjalin kemitraan dengan Dinas Kesehatan Kota Malang untuk memberikan peluang kepada mahasiswa atau peneliti lain melakukan penelitian di instansi terkait.
2. Bagi Dinas Kesehatan Kota Malang
 - a. Membantu Dinas Kesehatan dalam memanfaatkan data untuk memprediksi jumlah kasus DBD berdasarkan faktor lingkungan.
 - b. Memberikan kontribusi dalam membangun reputasi instansi sebagai sarana tempat yang mendukung penelitian mahasiswa.
3. Bagi Masyarakat
 - a. Memberikan informasi kepada masyarakat mengenai prediksi jumlah kasus DBD sehingga dapat lebih peduli terhadap upaya deteksi terjadinya DBD.
 - b. Meningkatkan kesadaran kepada masyarakat untuk lebih memahami faktor lingkungan untuk mencegah terjadinya DBD.

1.5 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penelitian ini yaitu:

1. Data yang digunakan terbatas pada data historis kasus DBD dan data iklim (curah hujan, suhu, kelembapan) di Kota Malang.
2. Penelitian ini tidak membahas aspek klinis dari DBD secara mendalam, melainkan berfokus pada prediksi jumlah kasus DBD berdasarkan data lingkungan.
3. Optimizer yang digunakan dalam *hyperparameter* GRU adalah *optimizer* Adam.

1.6 Definisi Istilah

Beberapa istilah yang digunakan dalam penelitian ini sebagai berikut:

- Backpropagation* : Algoritma pelatihan yang digunakan dalam jaringan saraf tiruan untuk mengoptimalkan parameter jaringan (seperti bobot dan *bias*) dengan cara meminimalkan fungsi kerugian (*loss function*).
- Batch Size* : Merujuk pada jumlah sampel data yang diproses dalam satu iterasi pelatihan (*batch*) sebelum parameter model diperbarui.
- Dropout* : Sebuah teknik regularisasi dalam jaringan saraf tiruan yang bertujuan untuk mencegah masalah *overfitting*. Teknik ini bekerja dengan menonaktifkan secara acak sejumlah unit beserta koneksinya selama pelatihan.

- Epoch* : Satu kali iterasi penuh di mana seluruh *dataset* telah, diproses dan digunakan untuk memperbarui parameter model, baik melalui proses maju (*forward pass*) maupun mundur (*backward pass*).
- Hyperparameter* : Parameter konfigurasi eksternal yang nilainya ditetapkan sebelum proses pelatihan dimulai. Dalam konteks GRU, ini mencakup Jumlah Unit, *Learning Rate*, *Dropout Rate*, dan *Batch Size*.
- Learning Rate* : Merupakan nilai skalar yang sangat krusial. Nilai ini menentukan seberapa besar langkah pembaruan bobot model pada tiap iterasi *gradient descent*.
- Min-Max Scaling* : Metode normalisasi data sederhana yang berfungsi untuk menyesuaikan dan mengubah nilai data ke dalam rentang kecil yang spesifik.
- Optimizer Adam* : Algoritma optimasi yang digunakan untuk memperbarui bobot model. Efektivitasnya baik untuk data besar dan masalah optimasi tidak stasioner, serta menawarkan konvergensi yang cepat dan stabil.
- Vanishing Gradient* : Fenomena yang terjadi selama pelatihan *deep learning* di mana gradien yang disebarkan ke lapisan-lapisan awal jaringan menjadi sangat kecil atau “menghilang”, sehingga menghambat kemampuan model untuk mempelajari dependensi dan hubungan jangka panjang.

BAB II

KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 *Preprocessing Data*

Tahapan *preprocessing* data dilakukan untuk memastikan data yang digunakan dalam pelatihan model GRU memiliki kualitas baik, bebas dari *noise*, dan siap digunakan dalam pemodelan deret waktu (Khairunisa & Hendikawati, 2024). Adapun langkah-langkah *preprocessing* adalah sebagai berikut:

1. Normalisasi Data: Seluruh fitur numerik dinormalisasi menggunakan metode *Min-Max Scaling* agar nilainya berada pada rentang 0–1, artinya *Min-Max Scaling* adalah sebuah proses transformasi nilai asli data ke dalam skala baru yang seragam, di mana nilai terkecil akan menjadi 0 dan nilai terbesar menjadi 1. Proses ini dilakukan untuk mempercepat konvergensi model serta memastikan setiap variabel memiliki kontribusi yang seimbang (Alkahfi dkk., 2024).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

Di mana x_{min} merupakan nilai paling rendah yang ditemukan pada satu fitur atau kolom tertentu di dalam dataset, sedangkan x_{max} merupakan nilai paling tinggi yang ditemukan pada fitur atau kolom tersebut.

2. Transformasi dan *Reshaping Data*: Setelah dinormalisasi, data diubah menjadi format input GRU berbentuk tensor tiga dimensi [*samples*, *timesteps*, *features*] menggunakan teknik *sliding window*. Teknik ini menggeser jendela berukuran tetap sepanjang deret waktu, sehingga nilai

dalam jendela menjadi input historis dan nilai berikutnya sebagai target prediksi. Proses ini mengubah data *time series* menjadi format yang sesuai untuk model GRU (Aksan dkk., 2023).

di mana:

x' : Hasil data setelah dinormalisasi

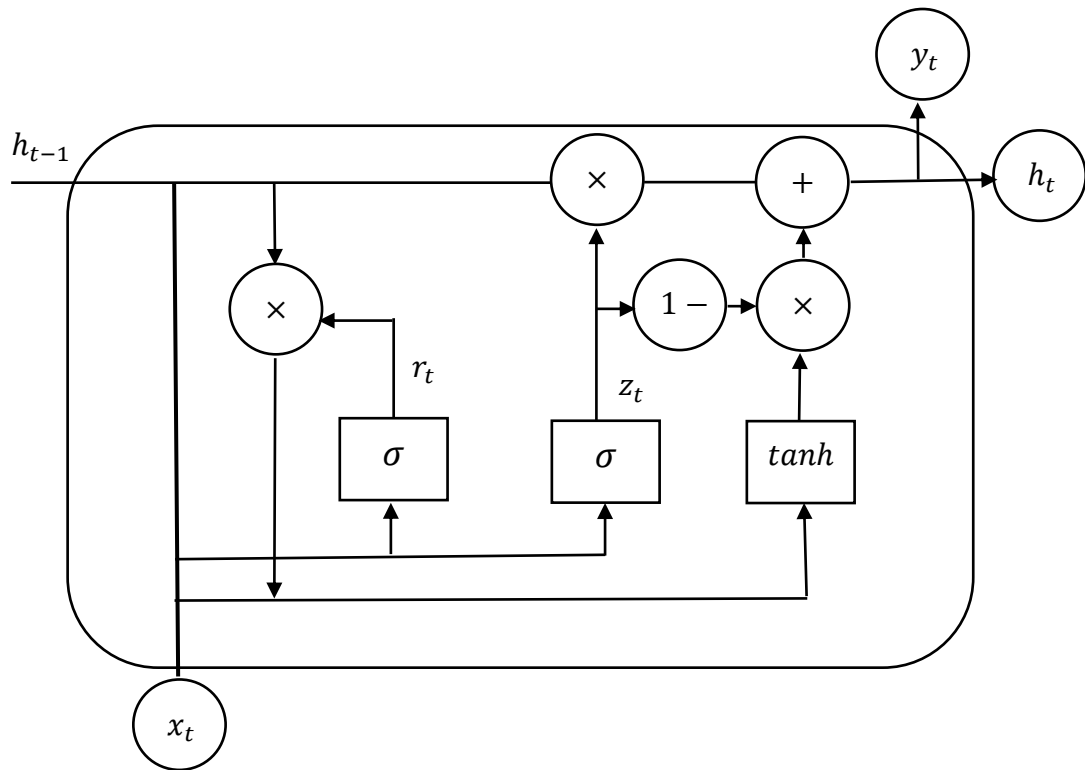
x : Data yang dinormalisasi

x_{min} : Nilai minimum dari keseluruhan data x

x_{max} : Nilai maksimum dari keseluruhan data x

2.1.2 *Gated Recurrent Unit (GRU)*

Gated Recurrent Unit (GRU) merupakan hasil pengembangan dari arsitektur *Recurrent Neural Network (RNN)* dan merupakan varian dari *Long Short-Term Memory (LSTM)* yang dirancang untuk mengatasi masalah *vanishing gradient*. Secara umum, GRU menerima input berupa vektor berdimensi (*timesteps, features*) dan menghasilkan output berupa prediksi berdasarkan seluruh urutan data yang diberikan. GRU menggunakan dua mekanisme utama berupa *update gate* dan *reset gate* yang berfungsi mengatur aliran informasi dari langkah waktu sebelumnya ke langkah waktu saat ini secara adaptif. GRU memiliki beberapa keunggulan signifikan, antara lain arsitekturnya yang lebih sederhana sehingga mengurangi kompleksitas komputasi, waktu pelatihan yang lebih cepat karena jumlah parameter yang lebih sedikit, serta performa yang kompetitif terutama dalam pengolahan data *time series* yang memerlukan pemahaman pola temporal secara efektif (Gema & Suhartono, 2020).



Gambar 2.1 Arsitektur GRU (Tu dkk., 2024)

Pada Gambar 2.1, notasi r_t sebagai *reset gate* yang menentukan sejauh mana informasi masa lalu harus dilupakan, sedangkan z_t sebagai *update gate* yang mengatur jumlah informasi dari *state* sebelumnya yang perlu dipertahankan ke *state* berikutnya. Sementara itu, σ dan $\tanh$ masing-masing menunjukkan *activation function* dan *candidate activation function* yang berperan sebagai komponen transformasi non-linear untuk memastikan aliran informasi di dalam sel tetap terkendali dan stabil (Oktaviani & Kharisudin, 2024). Dalam penelitian ini arsitektur yang digunakan ada pada Lampiran 2.

2.1.2.1 Mekanisme GRU

GRU memproses input berurutan melalui empat langkah utama. Pada setiap *timestep*, model menerima data masukan dan informasi dari *hidden state* sebelumnya untuk memperbarui informasi. Proses ini memungkinkan GRU

menangkap dan mempertahankan informasi penting dari periode sebelumnya tanpa mengalami masalah ketergantungan jangka panjang yang berlebihan. Penggunaan *gate* pada GRU mengontrol aliran informasi sehingga model dapat menentukan informasi mana yang perlu dipertahankan, diperbarui, atau diabaikan selama proses pembelajaran. Adapun mekanisme GRU sebagai berikut.

1. *Update Gate*

Membantu model untuk menentukan berapa banyak informasi masa lalu yang perlu diteruskan ke masa depan:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (2.2)$$

2. *Reset Gate*

Mengatur seberapa banyak informasi dari masa lalu yang dilupakan:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (2.3)$$

3. *Candidate Hidden State*

Menghasilkan calon informasi baru yang bisa dimasukkan ke dalam memori (*hidden state*) pada waktu ke- t :

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (2.4)$$

4. *Hidden State Update*

Menentukan berapa banyak informasi lama yang dipertahankan dan berapa banyak informasi baru dari *candidate* yang diambil:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (2.5)$$

5. *Output Layer*

Output *layer* berfungsi menghasilkan nilai prediksi berdasarkan informasi yang telah dipelajari dan disimpan pada *hidden state* (h_t).

$$\hat{y}_t = W_o h_t + b_o \quad (2.6)$$

di mana:

- x_t : Vektor input pada waktu (t)
- z_t : Vektor *update gate*
- r_t : Vektor *reset gate*
- h_t : Vektor *hidden state* saat ini
- $\tilde{h}_t$: Vektor *candidate hidden state*
- $\hat{y}_t$: Output atau nilai prediksi pada waktu (*timestep*) ke- t
- W_z : Matriks bobot input untuk *update gate*
- W_r : Matriks bobot input untuk *reset gate*
- W_h : Matriks bobot input untuk *candidate hidden state*
- W_o : Matriks bobot yang menghubungkan *hidden state* ke *layer output*
- U_z : Matriks bobot *recurrent* untuk *update gate* pada *hidden state* sebelumnya
- U_r : Matriks bobot *recurrent* untuk *reset gate* pada *hidden state* sebelumnya
- U_h : Matriks bobot *recurrent* untuk *candidate hidden state* pada *hidden state* sebelumnya
- b_z : Vektor bias untuk *update gate*
- b_r : Vektor bias untuk *reset gate*
- b_h : Vektor bias untuk *candidate hidden state*
- b_o : Vektor bias pada *layer output*
- σ : Fungsi aktivasi *sigmoid*
- $\tanh$: Fungsi aktivasi hiperbolik tangens
- $\odot$: Operasi perkalian elemen demi elemen

2.1.2.2 *Hyperparameter* GRU

Hyperparameter GRU menentukan struktur dan pelatihan model, memengaruhi dimensi dan perilaku pada persamaan (2.2) hingga (2.5) (Arfianti dkk., 2021). Untuk mendapatkan kombinasi parameter yang optimal, kombinasi ini menggunakan metode *grid search*, yaitu teknik optimasi yang bekerja dengan cara menguji secara sistematis seluruh kemungkinan kombinasi nilai *hyperparameter* yang telah ditentukan (Zhou dkk., 2025). Melalui evaluasi menyeluruh terhadap setiap kombinasi tersebut, *grid search* memastikan model terpilih memiliki performa terbaik berdasarkan metrik evaluasi yang ditetapkan (Dermawan dkk., 2023). *Hyperparameter* yang dioptimasi sebagai berikut:

1. Jumlah Unit (n)

Jumlah unit pada lapisan GRU menentukan dimensi vektor h_t , z_t , dan r_t , serta ukuran matriks bobot yang terlibat. Menurut penelitian, jumlah unit yang umum digunakan berkisar antara 32 hingga 256, dengan nilai 64 atau 128 sebagai pilihan *baseline* yang paling sering memberikan keseimbangan antara kapasitas model dan efisiensi komputasi (Bartz dkk., 2023).

2. Jumlah Lapisan

Jumlah lapisan pada arsitektur GRU menentukan kedalaman model dengan menumpuk beberapa lapisan GRU secara berurutan. Dalam praktik, penggunaan 1 hingga 2 lapisan GRU paling umum diterapkan karena mampu menangkap hubungan tanpa menyebabkan peningkatan risiko *vanishing gradient* atau *overfitting*. Penggunaan lebih dari 3

lapisan jarang dilakukan, kecuali data yang memiliki banyak faktor yang saling berkaitan (Bartz dkk., 2023).

3. *Learning Rate*

Learning rate mengontrol besarnya langkah pembaruan bobot selama proses optimasi dengan *gradient descent*. Nilai *learning rate* yang paling sering digunakan pada model GRU adalah 1×10^{-3} , dengan rentang umum antara 1×10^{-4} hingga 1×10^{-2} . Nilai yang terlalu tinggi dapat menyebabkan ketidakstabilan pelatihan, sedangkan nilai yang terlalu rendah butuh waktu yang sangat lama untuk mencapai hasil prediksi yang paling akurat (Zhang dkk., 2023).

4. *Dropout Rate*

Dropout rate menentukan proporsi unit yang diabaikan secara acak selama pelatihan untuk mencegah *overfitting*. Nilai *dropout rate* yang biasa digunakan pada model GRU berkisar antara 0.2 hingga 0.5, dengan nilai 0.2 atau 0.3 sebagai pilihan yang paling sering memberikan keseimbangan model tanpa mengurangi performa model secara berlebihan (Zhang dkk., 2023).

5. *Batch Size*

Batch size menentukan berapa sampel data yang akan digunakan untuk menghitung gradien dan memperbarui bobot model sebelum melakukan langkah pembaruan. Selama proses penelitian akan dilakukan beberapa percobaan nilai-nilai *batch size* untuk mengetahui nilai *batch size* mana yang akan memberikan hasil prediksi terbaik. Nilai seperti 32 atau 64

sering dipilih untuk menjaga keseimbangan antara stabilitas gradien dan efisiensi komputasi (Zhang dkk., 2023).

6. Jumlah *Epoch*

Jumlah *epoch* adalah frekuensi iterasi di mana seluruh dataset pelatihan telah melewati jaringan saraf secara lengkap baik pada tahap *forward pass*. Satu *epoch* didefinisikan sebagai satu siklus penuh pelatihan menggunakan seluruh sampel data. Penentuan jumlah *epoch* yang tepat sangat penting, nilai yang digunakan yaitu antara 50 hingga 200 umum digunakan untuk menghindari *underfitting* dan *overfitting*. Performa model dengan menerapkan teknik *early stopping* 10 sampai 20 untuk melihat *validation loss* dan menghentikan pelatihan secara otomatis ketika tidak ditemukan penurunan nilai kerugian yang lebih (Zhang dkk., 2023). Hal ini memastikan parameter yang disimpan adalah parameter yang menghasilkan kesalahan prediksi terkecil pada data validasi.

7. *Optimizer*

Algoritma pembaruan bobot seperti Adam banyak digunakan dalam pelatihan model *deep learning*. *Optimizer* Adam sangat efektif untuk optimasi masalah non-stasioner dan gradien yang jarang. Dibandingkan dengan algoritma optimasi lainnya, Adam memberikan konvergensi yang lebih cepat dan stabil, terutama pada model dengan jumlah parameter yang sangat banyak serta fungsi loss yang kompleks (Kingma & Ba, 2015). Fungsi *loss* yang digunakan dalam penelitian ini adalah *Mean Squared Error* (MSE), yang dirumuskan sebagai berikut:

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2 \quad (2.7)$$

Backpropagation merupakan algoritma utama untuk menghitung gradien fungsi *loss* terhadap seluruh parameter model secara efisien. Algoritma ini bekerja dengan menerapkan aturan rantai (*chain rule*) dari kalkulus secara mundur (*backward*) dari lapisan output menuju lapisan input. Melalui proses ini, kontribusi *error (loss)* terhadap setiap bobot dan bias dapat dihitung secara akurat, sehingga memungkinkan pembaruan parameter secara terarah (Rumelhart dkk., 1986). Proses perhitungan gradien *loss* terhadap parameter pada iterasi t dapat dinyatakan sebagai:

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial \theta} \quad (2.8)$$

Gradien yang diperoleh melalui *backpropagation* selanjutnya digunakan oleh *optimizer* Adam untuk memperbarui parameter model. Adapun rumus-rumus pada algoritma Adam adalah sebagai berikut:

- a. Estimasi Moment Pertama (*Mean*).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.9)$$

- b. Estimasi Moment Kedua (*Uncentered Variance*).

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (g_t)^2 \quad (2.10)$$

- c. Koreksi Bias (*Bias Correction*).

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \text{ dan } \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.11)$$

- d. Memperbarui Parameter (Bobot).

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (2.12)$$

di mana:

θ : Parameter yang akan dioptimalkan

η : Penentuan seberapa besar pembaruan bobot pada tiap iterasi

- g_t : Turunan dari *loss* terhadap parameter model
 m_t : Estimasi momen pertama (*mean*) dari gradien
 v_t : Estimasi momen kedua (variansi tak terpusat) dari gradien
 β_1, β_2 : Koefisien untuk eksponensial *moving average*
 ε : Nilai kecil untuk menghindari pembagian nol
 g_t : Gradien dari fungsi *loss* terhadap parameter model
 L : Fungsi *loss*

2.1.3 Kriteria Evaluasi

2.1.3.1 Mean Absolute Percentage Error (MAPE)

MAPE merupakan metrik evaluasi yang digunakan untuk mengukur akurasi model prediksi, MAPE menghitung rata-rata persentase kesalahan error antara nilai prediksi dan nilai aktual, dinyatakan dalam persen. Metrik ini sangat populer karena memberikan gambaran relatif tentang besarnya kesalahan dibandingkan dengan nilai aktual (Rodrigues & Machado, 2025)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \times 100\% \quad (2.13)$$

di mana:

- y_i : Nilai aktual
 $\hat{y}_t$: Nilai hasil prediksi
 n : Jumlah prediksi

Tabel 2.1 Kriteria Nilai MAPE

No	MAPE	Keterangan
1.	$MAPE < 10\%$	Sangat Baik
2.	$10\% \leq MAPE < 20\%$	Baik
3.	$20\% \leq MAPE < 50\%$	Cukup
4.	$MAPE \geq 50\%$	Buruk

2.1.4 Demam Berdarah Dengue (DBD)

Demam Berdarah Dengue (DBD) merupakan penyakit infeksi tropis yang disebabkan oleh Virus Dengue (DENV) dengan empat serotipe, yaitu DENV-1, DENV-2, DENV-3, dan DENV-4. Penyakit ini ditularkan melalui gigitan nyamuk *Aedes aegypti* yang terinfeksi (Marvianto dkk., 2023). DBD masih menjadi masalah kesehatan utama di Indonesia sebagai negara endemik dengue, terutama pada musim hujan. Secara global, sekitar 3.9 miliar penduduk tinggal di wilayah berisiko dengan estimasi 390 juta kasus setiap tahun, di mana sekitar 96 juta kasus menunjukkan gejala klinis. Di Indonesia, DBD menimbulkan beban kesehatan yang tinggi, terutama di wilayah perkotaan padat penduduk (Lestari dkk., 2023).

Infeksi dengue dapat berupa infeksi primer maupun sekunder. Infeksi primer umumnya menyebabkan demam dengue akut dengan gejala demam tinggi, nyeri otot dan sendi, sakit kepala, serta ruam kulit yang biasanya membaik dalam waktu sekitar tujuh hari (Simmons dkk., 2012). Sementara itu, infeksi sekunder dapat menyebabkan kondisi lebih berat seperti DBD atau sindrom renjatan dengue (SRD). Anak usia 3–14 tahun serta kelompok usia 15–39 tahun memiliki risiko lebih tinggi mengalami komplikasi berat dan renjatan (Nugraheni dkk., 2023).

Tingginya kasus DBD dipengaruhi oleh faktor seperti urbanisasi, perubahan iklim, rendahnya kesadaran masyarakat, serta kondisi lingkungan seperti curah hujan, suhu, kelembapan, dan genangan air yang mendukung perkembangbiakan *Aedes aegypti* (Rakhmatsani & Susanna, 2024). Oleh karena itu, diperlukan upaya pencegahan melalui edukasi masyarakat, vaksin dengue, dan penguatan sistem survei kesehatan (Schaefer dkk., 2021).

2.1.5 Curah Hujan

Curah hujan didefinisikan sebagai volume air hujan yang terkumpul pada permukaan datar tanpa adanya proses penguapan, perembesan ke dalam tanah, maupun aliran permukaan. Satuan yang digunakan adalah milimeter (mm), di mana 1 mm curah hujan berarti terdapat lapisan air setinggi 1 mm pada area seluas 1 m² (BMKG, 2017). Ketika intensitas hujan meningkat, air cenderung tergenang pada wadah alami (misalnya lubang pohon, daun yang menampung air) maupun wadah buatan yang terbuang di lingkungan terbuka (ban bekas, kaleng, pot bunga, dll.). Genangan-genangan tersebut menjadi tempat perindukan ideal bagi nyamuk *aedes aegypti*. Populasi nyamuk yang meningkat akibat bertambahnya lokasi perkembangbiakan berpotensi memperbesar risiko penularan virus dengue di wilayah tersebut (Lahdji & Putra, 2017).

Pengaruh curah hujan terhadap kepadatan nyamuk tidak selalu bersifat linier, melainkan dipengaruhi oleh berbagai faktor lingkungan yang saling berinteraksi. Faktor-faktor tersebut meliputi volume dan durasi hujan, kemampuan tanah dalam menyerap air, serta tingkat kelembapan udara yang cenderung meningkat setelah terjadinya hujan. Curah hujan dengan intensitas tertentu dapat menyebabkan terbentuknya genangan air pada berbagai wadah atau lingkungan sekitar yang berpotensi menjadi tempat berkembang biak nyamuk. Selain itu, peningkatan kelembapan udara pasca-hujan juga dapat mendukung kelangsungan hidup dan aktivitas nyamuk. Kombinasi dari faktor-faktor tersebut pada akhirnya dapat memperluas jumlah maupun variasi tempat perindukan nyamuk vektor sehingga berkontribusi terhadap peningkatan kepadatan populasi nyamuk.

2.1.6 Suhu Udara

Temperatur udara merupakan indikator derajat panas yang menunjukkan tingkat aktivitas gerakan molekul di atmosfer sekaligus mencerminkan panas atau dinginnya suatu benda atau lingkungan. Pengukuran suhu dilakukan menggunakan termometer dengan satuan skala Celsius ($^{\circ}\text{C}$), Reamur ($^{\circ}\text{R}$), atau Fahrenheit ($^{\circ}\text{F}$). Dalam pengamatan meteorologi konvensional, tercatat tiga nilai utama suhu udara, yaitu suhu tertinggi yang diperoleh dari termometer maksimum, suhu rata-rata yang dihitung berdasarkan pembacaan termometer bola kering dan bola basah, serta suhu terendah yang didapat dari termometer minimum (Kahar dkk., 2024).

Nyamuk masih mampu bertahan pada temperatur rendah, namun ketika suhu lingkungan turun di bawah ambang kritis, laju metabolisme mereka akan menurun drastis atau bahkan terhenti sama sekali. Kisaran suhu optimal untuk perkembangan dan aktivitas nyamuk adalah $25\text{--}27^{\circ}\text{C}$. Apabila suhu udara mencapai di atas 40°C atau di bawah 10°C , proses pertumbuhan dan reproduksi nyamuk akan terhenti total. Menurut Sumantri (2017), temperatur udara memberikan pengaruh signifikan terhadap beberapa aspek biologis nyamuk, yaitu, laju metabolisme yang paling aktif pada rentang $32\text{--}35^{\circ}\text{C}$, sedangkan suhu yang terlalu tinggi justru memperlambat fungsi fisiologis, perkembangan stadia (telur, larva, pupa, hingga dewasa) mencapai kondisi paling baik pada suhu $25\text{--}27^{\circ}\text{C}$, dan siklus gonotrofik (proses pencarian inang dan bertelur), serta masa hidup nyamuk yang cenderung lebih singkat ketika suhu lingkungan berada di atas $27\text{--}30^{\circ}\text{C}$ (Agustin dkk., 2017).

2.1.7 Kelembapan Udara

Kelembapan udara merupakan jumlah uap air yang terkandung dalam atmosfer, yang dapat diukur melalui tiga parameter utama, yaitu kelembapan mutlak (jumlah uap air aktual per satuan volume udara), kelembapan relatif (rasio antara tekanan uap air aktual dengan tekanan uap air pada kondisi jenuh), serta defisit tekanan uap (Umar dkk., 2022).

Pengukuran kelembapan udara dilakukan dengan alat higrometer dan dinyatakan dalam satuan persen (%) (Rachmadiyanto, 2018). Tingkat kelembapan yang terlalu tinggi menyebabkan permukaan bangunan menjadi lembap dan berair, sehingga menciptakan lingkungan yang sangat mendukung pertumbuhan mikroorganisme patogen seperti bakteri dan jamur. Selain itu, kelembapan relatif turut memengaruhi penyebaran serta durasi siklus hidup nyamuk vektor; wilayah dengan vegetasi lebat dan kelembapan tinggi cenderung lebih rentan terhadap fluktuasi kelembapan dibandingkan daerah kering (Makruf dkk., 2025).

2.2 Kajian Integrasi Topik dengan Al-Quran/Hadits

Perkembangan ilmu pengetahuan dan teknologi pada era modern telah memberikan ruang luas bagi manusia untuk memahami berbagai fenomena alam yang terjadi di sekitar. Dalam bidang kesehatan masyarakat, salah satu tantangan yang masih terus dihadapi adalah penyebaran penyakit menular seperti Demam Berdarah Dengue (DBD). Penyakit ini memiliki karakteristik yang erat kaitannya dengan faktor lingkungan, iklim, serta perilaku manusia, sehingga analisis terhadap data yang menggambarkan kondisi-kondisi tersebut menjadi penting. Salah satu pendekatan yang banyak digunakan adalah pemanfaatan metode kecerdasan buatan

seperti *Gated Recurrent Unit* (GRU), yang memiliki kemampuan dalam menganalisis data deret waktu (*time series*) dan menghasilkan prediksi terhadap potensi peningkatan kasus DBD di masa mendatang. Upaya ini tidak hanya memiliki nilai praktis dalam bidang kesehatan, tetapi juga mencerminkan penerapan akal dan ilmu sebagaimana dianjurkan ajaran Islam. Salah satunya dalam firman Allah SWT QS. Al-Baqarah [2]: 164 (Kemenag, 2024), sebagai berikut.

إِنَّ فِي خَلْقِ السَّمُوتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ وَالْفُلْكِ الَّتِي تَجْرِي فِي الْبَحْرِ بِمَا يَنْفَعُ النَّاسَ وَمَا أَنْزَلَ اللَّهُ مِنَ السَّمَاءِ مِنْ مَّاءٍ فَأَحْيَا بِهِ الْأَرْضَ بَعْدَ مَوْتِهَا وَبَثَّ فِيهَا مِنْ كُلِّ دَابَّةٍ وَتَصْرِيفِ الرِّيْحِ وَالسَّحَابِ الْمُسَخَّرِ بَيْنَ السَّمَاءِ وَالْأَرْضِ لَآيَاتٍ لِّقَوْمٍ يَعْقِلُونَ ﴿١٦٤﴾

“Sesungguhnya pada penciptaan langit dan bumi, pergantian malam dan siang bahtera yang berlayar di laut dengan (muatan) yang bermanfaat bagi manusia, apa yang Allah turunkan dari langit berupa air, lalu dengannya Dia menghidupkan bumi setelah mati (kering), dan Dia menebarkan di dalamnya semua jenis hewan, dan pengisaran angin dan awan yang dikendalikan antara langit dan bumi, (semua itu) sungguh merupakan tanda-tanda (kebesaran Allah) bagi kaum yang mengerti.” (QS. Al-Baqarah [2]: 164)

Ayat tersebut mengandung perubahan fenomena alam seperti hujan, angin, dan pergantian musim merupakan tanda-tanda kekuasaan Allah yang menunjukkan keteraturan sistem ciptaan-Nya. Dalam Al-Qur'an, Allah berulang kali mengingatkan manusia untuk memperhatikan dan merenungkan gejala-gejala alam sebagai bentuk pembelajaran dan penguatan keimanan. Setiap unsur cuaca yang terjadi di bumi bukan sekadar fenomena fisik, melainkan memiliki hikmah dan fungsi yang dapat dipahami melalui pendekatan ilmiah. Oleh karena itu, pengamatan terhadap dinamika alam menjadi bagian dari perintah untuk berpikir kritis dan mendalam terhadap ayat-ayat kauniyah yaitu tanda kebesaran Allah.

Dalam konteks penelitian modern, perintah untuk memperhatikan alam ini dapat dimaknai sebagai dorongan bagi manusia untuk melakukan kajian ilmiah terhadap berbagai fenomena yang terjadi di lingkungan sekitarnya. Salah satu contohnya adalah mempelajari keterkaitan antara faktor cuaca seperti curah hujan, suhu, dan kelembapan, dengan peningkatan kasus penyakit menular seperti Demam Berdarah Dengue (DBD). Pendekatan ini menunjukkan bahwa memahami pola cuaca tidak hanya secara ilmiah, tetapi juga bentuk tanggung jawab manusia dalam menjaga lingkungan dan kesehatan masyarakat. Dengan demikian, ilmu pengetahuan menjadi sarana untuk mengoptimalkan potensi alam demi kebaikan.

Selain itu, penelitian mengenai hubungan antara cuaca dan penyakit dapat dipandang sebagai wujud nyata dari amanah khalifah yang diberikan Allah kepada manusia untuk memelihara bumi. Islam mengajarkan pentingnya menjaga kebersihan, kesehatan, dan keseimbangan ekosistem sebagai bagian dari ibadah dan rasa syukur atas nikmat yang telah diberikan. Upaya memahami dan memprediksi pola penyebaran DBD berdasarkan kondisi cuaca merupakan langkah yang tidak hanya berlandaskan ilmu pengetahuan, tetapi juga sejalan dengan nilai-nilai keislaman dalam menjaga kehidupan dan mencegah mudarat.

Dengan demikian, penelitian ini memiliki dimensi spiritual dan sosial. Melalui pemanfaatan ilmu pengetahuan seperti analisis cuaca dan penerapan model prediksi berbasis teknologi, Hal ini mencerminkan sinergi antara sains dan iman, di mana pengetahuan yang diperoleh dari pengamatan terhadap alam menjadi sarana untuk meningkatkan kesejahteraan manusia serta memperkuat kesadaran akan kebesaran Sang Pencipta.

2.3 Kajian Topik dengan Teori Pendukung

Analisis prediksi insiden Demam Berdarah Dengue (DBD) menggunakan model *Gated Recurrent Unit* (GRU) diawali dengan tahap pengumpulan dataset secara sistematis. Dataset yang digunakan terdiri atas data *time series* kasus DBD serta data iklim yang meliputi curah hujan, suhu rata-rata, dan kelembapan udara. Semua data diperoleh dari lembaga resmi supaya menjamin validitas sumber.

Tahap selanjutnya adalah *preprocessing* data, yang bertujuan untuk meningkatkan kualitas data sebelum dimasukkan ke dalam model. Proses ini meliputi normalisasi menggunakan metode *Min-Max Scaling* untuk menyamakan rentang nilai seluruh fitur ke dalam skala yang seragam. Normalisasi ini berfungsi mempercepat proses konvergensi model serta meningkatkan stabilitas pembelajaran jaringan saraf. Selanjutnya, data yang telah dinormalisasi ditransformasi ke dalam format khusus yang sesuai dengan input GRU, yaitu *[samples, timesteps, features]*. Misalnya, apabila digunakan enam bulan sebagai *timesteps* dan empat fitur (kasus DBD, curah hujan, suhu udara, kelembapan udara), maka data diubah menjadi format $[n, 6, 4]$, dengan n merupakan jumlah sampel.

Tahap berikutnya adalah inisialisasi *hyperparameter* yang berfungsi mengatur proses pembelajaran model GRU. Beberapa *hyperparameter* utama yang digunakan meliputi jumlah unit, jumlah lapisan, *learning rate*, *batch size*, jumlah *epoch*, dan *dropout rate*. Nilai-nilai *hyperparameter* ini ditentukan melalui proses optimasi menggunakan metode *grid search* atau *random search* untuk memperoleh kombinasi dengan *error* terkecil pada data pelatihan.

Setelah *hyperparameter* optimal ditentukan, dilakukan pembangunan arsitektur model GRU. Model GRU dirancang untuk menerima input berupa urutan data *time series* dan dibangun berdasarkan mekanisme utama yang terdiri dari *update gate*, *reset gate*, *candidate hidden state*, dan *hidden state update*. Tahap ini memastikan model mampu menangkap hubungan temporal antar variabel input dengan efektif.

Selanjutnya dilakukan pelatihan model (*training*) menggunakan 80% data sebagai *data training* dan 20% sebagai *data testing*. Pada tahap ini, model mempelajari pola hubungan antara variabel iklim dan jumlah kasus DBD. Proses pelatihan melibatkan langkah *forward pass* untuk menghasilkan prediksi awal, kemudian menghitung selisih antara hasil prediksi dan nilai aktual menggunakan *Mean Squared Error* (MSE) sebagai *loss function*. Nilai *loss* yang dihasilkan digunakan untuk memperbarui bobot jaringan melalui bantuan *Adam Optimizer* yang memanfaatkan prinsip *adaptive learning rate* dan *momentum*. Pelatihan dilakukan secara iteratif selama beberapa *epoch* hingga nilai *loss* menunjukkan konvergensi.

Tahap selanjutnya adalah pengujian model (*testing*) menggunakan data yang tidak digunakan selama pelatihan. Model yang telah terlatih menerima input dari *data testing* dan menghasilkan nilai prediksi jumlah kasus DBD. Hasil prediksi tersebut kemudian dikembalikan ke skala aslinya melalui proses denormalisasi, dan dibandingkan dengan nilai aktual menggunakan metrik evaluasi *Mean Absolute Percentage Error* (MAPE). Nilai MAPE yang rendah menunjukkan bahwa model memiliki kemampuan prediksi yang baik terhadap data baru.

Hasil pelatihan dan pengujian divisualisasikan dalam bentuk grafik perbandingan antara nilai aktual dan hasil prediksi. Visualisasi ini bertujuan untuk menunjukkan kemampuan model GRU dalam mengikuti pola tren insiden DBD pada periode tertentu. Dengan rangkaian tahap *training*, *optimasi*, dan *testing* yang dilakukan secara sistematis, model GRU yang dikembangkan diharapkan memiliki performa yang akurat terhadap data baru.

Sebagai hasil akhir, kemudian berlanjut pada tahap *forecasting* (peramalan), di mana model yang telah mempelajari pola dan ketergantungan temporal tersebut digunakan untuk memproyeksikan nilai di masa depan. Keunggulan struktur GRU yang lebih sederhana namun efisien menjadikannya metode yang sesuai untuk menghasilkan estimasi fluktuasi kasus sebagai bentuk kewaspadaan dini dan sistem peringatan bagi masyarakat dalam upaya pengendalian wabah secara lebih efektif.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian ini merupakan penelitian kuantitatif dengan pendekatan prediksi berbasis *machine learning*, khususnya menggunakan algoritma *Gated Recurrent Unit* (GRU) untuk memprediksi jumlah kasus Demam Berdarah Dengue (DBD). Penelitian ini bersifat non-eksperimental, dengan fokus pada pengembangan dan evaluasi model GRU untuk memprediksi jumlah kasus DBD berdasarkan data deret waktu (*time series*). Pendekatan ini melibatkan analisis data sekunder, seperti data kasus DBD dan faktor iklim, untuk menghasilkan prediksi yang akurat.

3.2 Data dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder berupa data deret waktu (*time series*) yang mencakup jumlah kasus DBD dan faktor-faktor lingkungan yang memengaruhinya, dikumpulkan untuk periode tertentu, yaitu Januari 2015 hingga Oktober 2025 dengan 130 data. Data kasus DBD, diperoleh dari laporan resmi Dinas Kesehatan Kota Malang. Data iklim bersumber dari *website* Badan Meteorologi, Klimatologi, dan Geofisika (BMKG).

3.3 Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui studi dokumentasi dengan mengambil data sekunder dari instansi terkait guna menjamin keakuratan informasi. Data

jumlah kasus DBD diperoleh melalui permohonan resmi ke Dinas Kesehatan Kota Malang, sedangkan data iklim diunduh langsung dari situs resmi BMKG. Untuk menjamin kualitas, dilakukan proses pemeriksaan ulang (verifikasi) dengan mencocokkan angka yang telah dicatat dengan dokumen asli. Selanjutnya, data diolah dengan menyusunnya ke dalam format tabel bulanan yang sistematis agar siap digunakan untuk tahap analisis.

3.4 Variabel Penelitian

Variabel yang dianalisis dalam penelitian ini terdiri dari dua kategori utama, yaitu variabel dependen (Y) dan variabel independen (X). Variabel dependen adalah hasil yang ingin diprediksi, sedangkan variabel independen adalah faktor-faktor yang memengaruhi hasil tersebut. Rincian variabel yang digunakan adalah sebagai berikut.

Tabel 3.1 Variabel-variabel Penelitian

Variabel	Sumber Data	Bentuk Data	Proses Pengolahan	Keterangan
<i>Lag</i> Kasus DBD	Dinas Kesehatan Kota Malang	Bulanan	Data kasus DBD ditransformasi menggunakan metode <i>time-shift</i> untuk membentuk variabel <i>lag</i> kasus DBD. Selanjutnya, data disusun menjadi sekuensial dengan panjang <i>timestep</i> 6 dan dinormalisasi dengan <i>Min-Max Scaling</i> sebelum digunakan sebagai input GRU.	X_1

Lanjutan Tabel 3.1 Variabel-variabel Penelitian

Variabel	Sumber Data	Bentuk Data	Proses Pengolahan	Keterangan
Curah Hujan	Badan Meteorologi, Klimatologi, dan Geofisika (BMKG)	Harian (mm)	Data harian diubah menjadi data bulanan dengan menghitung total dan rata-rata per bulan. Selanjutnya normalisasi <i>Min Max Scaling</i> sebelum digunakan sebagai input sekuensial GRU.	X_2
Suhu Udara	Badan Meteorologi, Klimatologi, dan Geofisika (BMKG)	Harian ($^{\circ}\text{C}$)	Data harian diubah menjadi data bulanan dengan menghitung total dan rata-rata suhu udara per bulan. Selanjutnya normalisasi <i>Min Max Scaling</i> sebelum digunakan sebagai input sekuensial GRU.	X_3
Kelembapan Udara	Badan Meteorologi, Klimatologi, dan Geofisika (BMKG)	Harian (%)	Data harian diubah menjadi data bulanan dengan menghitung total dan rata-rata kelembapan udara per bulan. Selanjutnya normalisasi <i>Min Max Scaling</i> sebelum digunakan input sekuensial GRU.	X_4
Jumlah Kasus DBD	Dinas Kesehatan Kota Malang	Bulanan	Data digunakan sebagai target. Sebelum pelatihan, data diselaraskan dengan periode waktu variabel cuaca. Nilai Y juga dapat diskalakan (<i>Min Max Scaling</i>) agar sebanding dengan input GRU.	Y

3.5 Teknik Analisis Data

Alur penelitian dirancang secara sistematis untuk menerapkan metode *Gated Recurrent Unit* (GRU) dalam memprediksi jumlah kasus Demam Berdarah Dengue (DBD). Langkah-langkah penelitian adalah sebagai berikut:

1. Pengumpulan Dataset

Mengumpulkan dataset insiden DBD dan data iklim dari sumber resmi.

2. Pembagian Dataset

Proses analisis diawali dengan pembagian dataset ke dalam dua subset berbeda yang berfungsi untuk pelatihan dan pengujian model. Sebagian besar data, sekitar 80%, digunakan sebagai *training set* agar model dapat mempelajari pola dan karakteristik dari data historis. Sementara itu, sisanya, sekitar 20%, dijadikan *testing set* yang tidak dilibatkan selama proses pelatihan dan digunakan untuk mengevaluasi kemampuan model dalam memprediksi data baru. Pembagian dilakukan secara berurutan berdasarkan waktu (data awal sebagai *training*, data terbaru sebagai *testing*), tanpa pengacakan untuk menjaga karakteristik *time series* dan menghindari data *leakage*.

3. *Preprocessing* Data

Data diolah untuk mengatasi masalah kualitas data dan mempersiapkannya untuk model GRU. Langkah-langkahnya meliputi:

- a. Normalisasi Fitur Numerik: Fitur numerik dinormalisasi menggunakan *Min-Max Scaling* seperti (2.1).
- b. *Reshaping* Data: Mengubah data menjadi format [*samples*, *timesteps*, *features*] untuk input GRU. Misalnya, menggunakan 6 bulan sebagai

timesteps dan 4 variabel (kasus DBD, curah hujan, suhu udara, kelembapan udara), data di-*reshape* menjadi $[n, 6, 4]$, di mana n adalah jumlah sampel.

4. Inisialisasi *Hyperparameter*

Hyperparameter ditentukan sebelum proses pelatihan dimulai dan tidak berubah selama pelatihan berlangsung. *Hyperparameter* yang diinisialisasi meliputi jumlah unit GRU, adapun kombinasi *hyperparameter* sebagai berikut.

Tabel 3.2 Kombinasi *Hyperparameter*

<i>Hyperparameter</i>	Nilai yang Diuji	Jumlah
Jumlah Unit	32, 64	2
Jumlah Lapisan	1, 2	2
<i>Learning Rate</i>	0.01, 0.001	2
<i>Dropout Rate</i>	0.1, 0.2	2
<i>Epoch</i>	100, 200	2
<i>Batch Size</i>	8, 16	2
Total		64

5. Pembangunan Model GRU

Model GRU dirancang untuk menerima masukan berupa urutan data *time series*. Arsitektur GRU dibangun berdasarkan persamaan matematis utama, sebagai berikut:

- Update gate* pada persamaan (2.2)
- Reset gate* pada persamaan (2.3)
- Candidate hidden state* pada persamaan (2.4)
- Hidden state update* pada persamaan (2.5)
- Output Layer* pada persamaan (2.6)

6. Pelatihan dan Optimasi Model

Pada tahap ini, model dilatih untuk menyesuaikan bobot dan bias agar mampu meminimalkan nilai *error*. Proses dilakukan secara berulang menggunakan

metode *grid search* untuk menemukan kombinasi hyperparameter terbaik dari Tabel 3.2. Proses pada setiap iterasi meliputi:

- a. *Forward propagation*, yaitu data diproses melalui model GRU untuk menghasilkan nilai prediksi.
- b. Perhitungan *loss function* menggunakan *Mean Squared Error* (MSE) pada persamaan (2.7).
- c. *Backpropagation* untuk menghitung gradien *error* terhadap parameter model pada persamaan (2.8).
- d. Pembaruan bobot menggunakan *optimizer* Adam berdasarkan gradien yang diperoleh dari *backpropagation* pada persamaan (2.9) sampai (2.12).
- e. Proses diulang hingga mencapai jumlah *epoch* yang ditentukan.

7. Pemilihan Konfigurasi Terbaik

Setelah seluruh kombinasi *hyperparameter* diuji, dipilih model dengan nilai *error* terendah sebagai model final untuk tahap pengujian.

8. Pengujian Model (*Testing Process*)

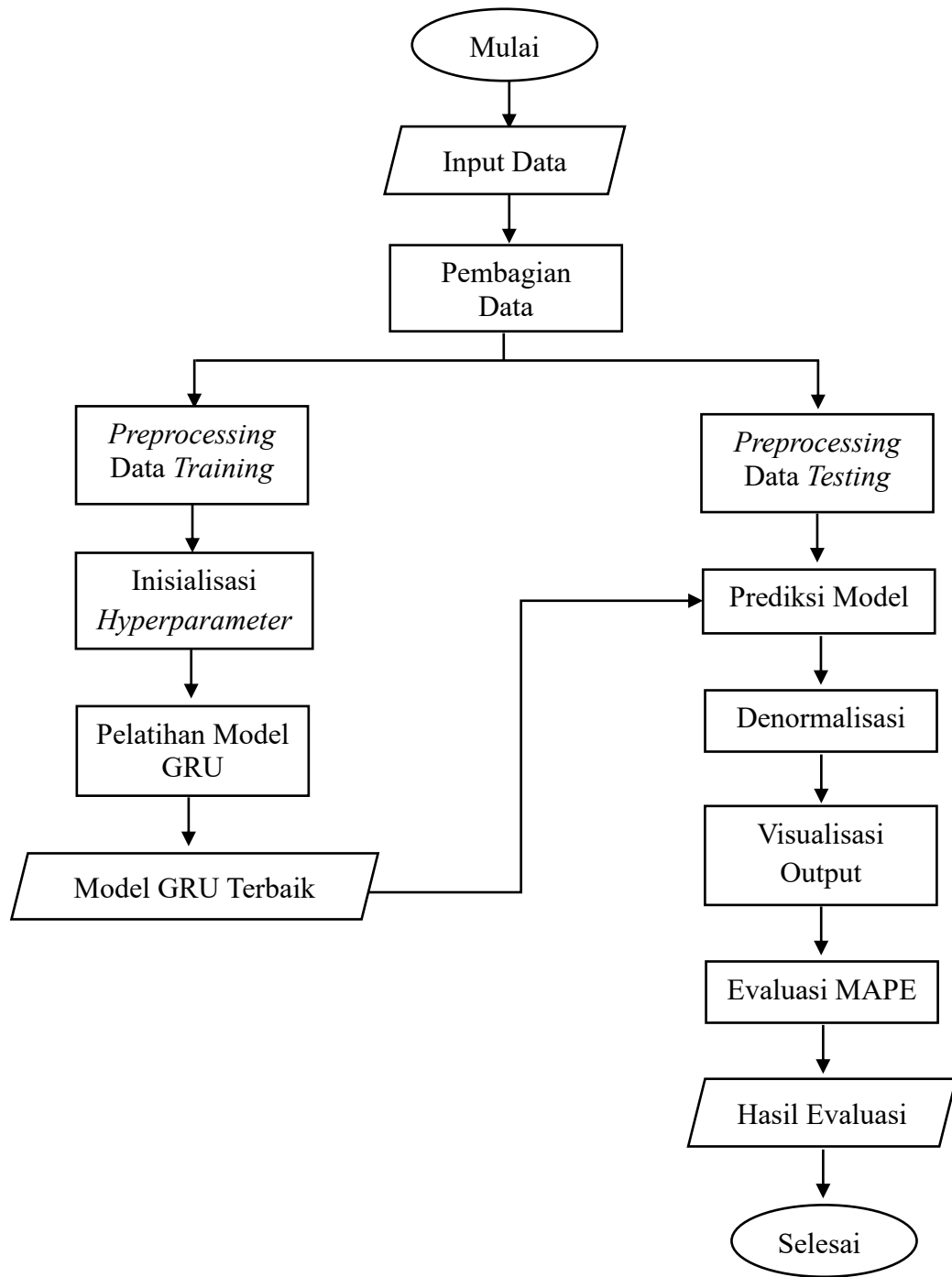
Model final yang telah terpilih diuji menggunakan data *testing* (20%) yang sebelumnya telah dipisahkan. Langkah-langkahnya meliputi:

- a. Denormalisasi: Mengembalikan nilai prediksi ke skala asli.
- b. Evaluasi Metrik: Menghitung tingkat akurasi model menggunakan metrik *Mean Absolute Percentage Error* (MAPE) pada persamaan (2.13).

9. Visualisasi dan Interpretasi

Hasil pengujian disajikan dalam bentuk grafik perbandingan antara data aktual dan data prediksi. Tahap ini bertujuan untuk menilai kemampuan model GRU dalam mengikuti pola fluktuasi kasus DBD secara visual.

3.6 Diagram Alir Penelitian



Gambar 3.1 Diagram Alir Penelitian

BAB IV

HASIL DAN PEMBAHASAN

4.1 Statistika Deskriptif

Data observasi yang digunakan merupakan data bulanan selama periode pengamatan yang mencakup variabel kasus DBD, curah hujan, suhu udara, dan kelembapan udara. Data tersebut dianalisis menggunakan bantuan perangkat lunak pengolah data untuk memperoleh nilai minimum, maksimum, *mean*, median, dan varians. Hasil perhitungan statistika deskriptif disajikan pada Tabel 4.1.

Tabel 4.1 Statistika Deskriptif Data

Variabel	<i>Mean</i>	Median	Varians	Min	Max
Kasus DBD	38.26	17.50	1593.80	1.00	155.00
Curah Hujan (mm)	360.78	304.08	136211.71	0.00	1712.34
Suhu Udara (°C)	24.21	24.20	0.8341	21.94	26.49
Kelembapan Udara (%)	78.43	78.80	24.56	62.03	87.03

Keterangan: Varians memiliki satuan kuadrat dari satuan variabel, yaitu mm², °C², dan %².

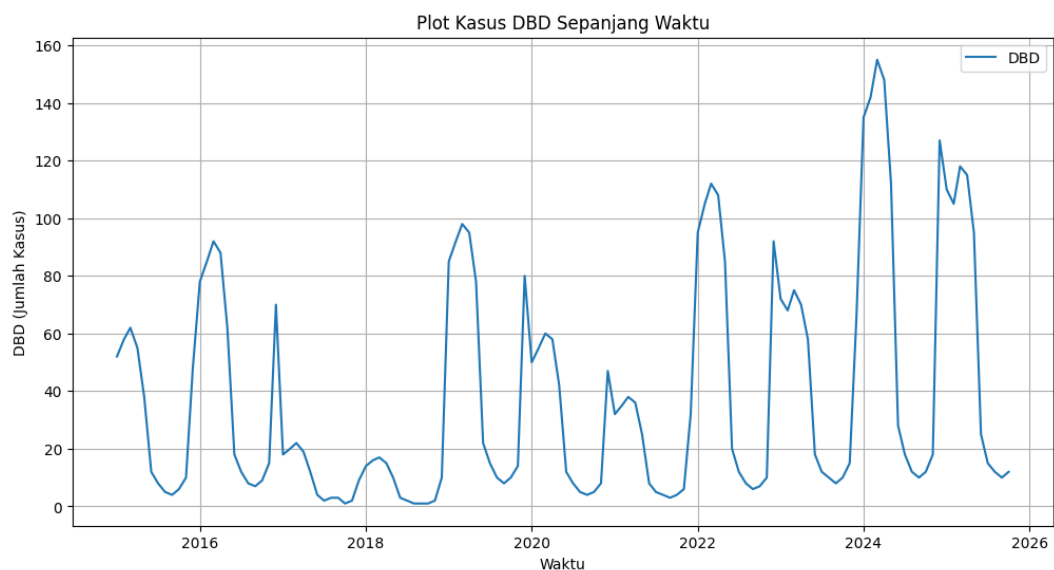
Berdasarkan Tabel 4.1, kasus DBD memiliki nilai *mean* sebesar 38.26 atau 39 kasus dan median sebesar 17.50 atau 18 kasus. Perbedaan yang cukup jauh antara *mean* dan median menunjukkan bahwa distribusi data DBD cenderung tidak simetris, dengan adanya beberapa nilai yang relatif tinggi sehingga meningkatkan nilai rata-rata. Hal ini didukung oleh nilai varians sebesar 1593.80 kasus² yang tergolong besar, menandakan bahwa data kasus DBD memiliki tingkat keragaman yang tinggi dan berfluktuasi secara signifikan.

Curah hujan memiliki nilai *mean* sebesar 360.78 mm dan median sebesar 304.08 mm. Nilai *mean* yang lebih besar dibandingkan median menunjukkan adanya kecenderungan distribusi miring ke kanan (*right-skewed*), di mana terdapat beberapa nilai curah hujan yang jauh lebih besar dibandingkan sebagian besar data.

Hal ini terlihat dari nilai maksimum yang mencapai 1712.34 mm, yang sangat jauh di atas nilai rata-rata. Selain itu, nilai varians yang sangat besar, yaitu 136211.71 mm², menunjukkan bahwa data memiliki penyebaran yang sangat luas..

Pada variabel suhu udara, nilai *mean* sebesar 24.21 °C dan median sebesar 24.20 °C menunjukkan bahwa data suhu udara relatif simetris dan terpusat di sekitar nilai tengah. Selisih yang sangat kecil antara *mean* dan median mengindikasikan kestabilan suhu udara. Hal ini diperkuat oleh nilai varians yang rendah, yaitu sebesar 0.834 °C², yang menandakan bahwa variasi suhu udara relatif kecil.

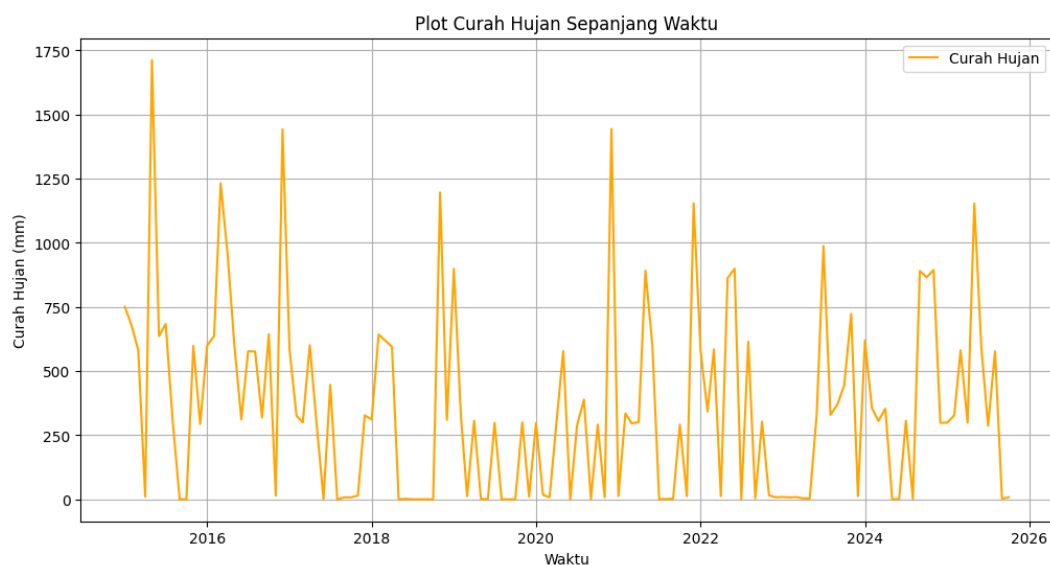
Sementara itu, variabel kelembapan udara memiliki nilai *mean* sebesar 78.43% dan median sebesar 78.80%, yang menunjukkan bahwa distribusi data kelembapan udara cukup seimbang. Nilai varians sebesar 24.56%² mengindikasikan bahwa kelembapan udara memiliki variasi yang relatif kecil dan cenderung stabil di sekitar nilai rata-ratanya.



Gambar 4.1 Plot Data Kasus DBD

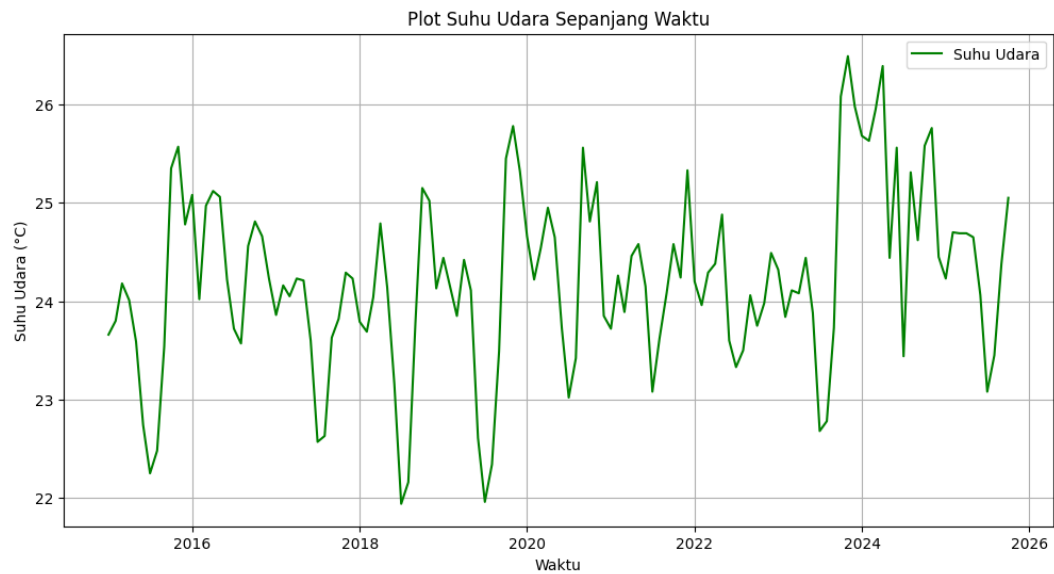
Berdasarkan Gambar 4.1 jumlah kasus DBD menunjukkan pola fluktuasi musiman yang konsisten dengan tren yang cenderung meningkat pada tahun-tahun terakhir. Penurunan kasus yang signifikan (mencapai titik terendah) secara rutin

terjadi pada pertengahan hingga akhir tahun, seperti yang terlihat jelas pada periode pertengahan 2017, 2018, dan 2021. Sebaliknya, lonjakan kasus atau puncak epidemi terjadi pada awal setiap tahunnya. Puncak kasus tertinggi dalam satu dekade terakhir tercatat terjadi pada kuartal pertama tahun 2024, di mana jumlah penderita melonjak tajam hingga melebihi 150 kasus, diikuti oleh lonjakan besar lainnya pada awal tahun 2025 yang mencapai angka di atas 120 kasus.



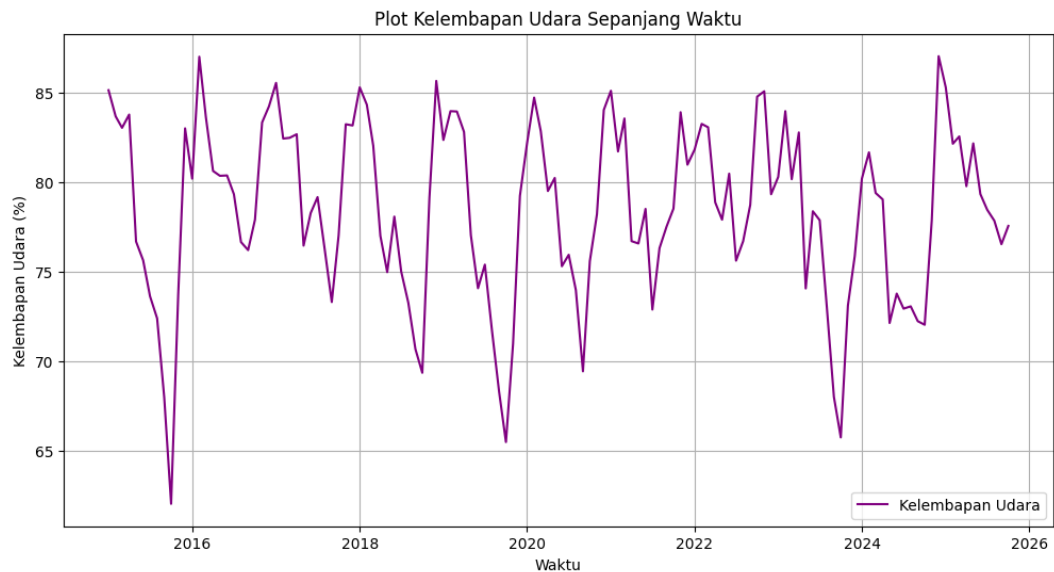
Gambar 4.2 Plot Data Curah Hujan

Berdasarkan Gambar 4.2 curah hujan menunjukkan intensitas yang sangat variatif dengan pola curah hujan ekstrem yang muncul secara periodik. Curah hujan tertinggi tercatat pada awal tahun 2015 dengan intensitas melebihi 1.700mm. Selama rentang pengamatan, terdapat beberapa periode dengan intensitas hujan yang sangat tinggi (di atas 1.000mm), yaitu pada awal tahun 2016, awal 2017, awal 2021, dan awal 2022. Di sisi lain, wilayah ini juga mengalami periode kering yang ekstrem di mana curah hujan berada di titik nol atau mendekati nol dalam durasi yang cukup lama, terutama yang terlihat jelas pada pertengahan hingga akhir tahun 2015, 2018, 2019, 2023, dan 2024.



Gambar 4.3 Plot Data Suhu Udara

Berdasarkan Gambar 4.3 menunjukkan fluktuasi dinamis dalam rentang nilai 22°C hingga 26.5°C. Penurunan suhu yang signifikan hingga mencapai titik terendah di bawah 22.5°C tercatat terjadi pada pertengahan tahun 2015, 2018, dan 2019, yang mengindikasikan adanya periode udara yang lebih dingin secara periodik. Sebaliknya, peningkatan suhu yang cukup tajam mulai terlihat pada akhir tahun 2015 dan 2019, hingga mencapai puncaknya pada sepanjang tahun 2024 di mana suhu udara secara konsisten berada pada level tertinggi di atas 25.5°C. Secara keseluruhan, variabel suhu udara ini menunjukkan tren yang sedikit meningkat di akhir periode pengamatan, khususnya pada tahun 2024 hingga 2025, yang menciptakan lingkungan hangat bagi aktivitas vektor penyakit.



Gambar 4.4 Plot Data Kelembapan Udara

Berdasarkan Gambar 4.4 menggambarkan variabel kelembapan udara yang memiliki pola musiman yang sangat teratur dan konsisten setiap tahunnya. Tingkat kelembapan udara mencapai puncaknya di atas 85% pada awal tahun, seperti yang terlihat jelas pada awal periode 2016, 2018, 2020, hingga mencapai titik tertinggi pada awal tahun 2025. Di sisi lain, penurunan kelembapan yang drastis di bawah 70% terjadi secara rutin pada pertengahan hingga akhir tahun, dengan titik terendah ekstrem tercatat pada periode 2015, 2019, dan 2023. Pola gelombang yang stabil pada grafik kelembapan ini mencerminkan siklus tahunan yang kontras antara musim penghujan dan musim kemarau, di mana tingginya kelembapan pada awal tahun secara visual sangat selaras dengan periode peningkatan kasus DBD yang telah dianalisis sebelumnya.

4.2 Preprocessing Data

Data *time series* yang telah disusun secara kronologis terlebih dahulu dibagi menjadi dua bagian, yaitu 80% sebagai data latih yang berjumlah 104 data dan 20% sebagai data uji yang berjumlah 26 data. Selanjutnya, dilakukan tahap pra-

pengolahan (*preprocessing*) pada masing-masing bagian data yang meliputi normalisasi untuk menyamakan rentang nilai dan perubahan bentuk (*reshaping*) agar data sesuai dengan format input yang dibutuhkan oleh model GRU.

4.2.1 Normalisasi Data

Normalisasi data dilakukan menggunakan metode *Min-Max Scaling*, yaitu dengan melakukan transformasi nilai setiap variabel ke dalam rentang 0 hingga 1. Metode ini dipilih karena mampu mempertahankan pola distribusi data asli serta sesuai untuk pemodelan data runtun waktu. Secara matematis, proses ini dilakukan dengan mengurangkan nilai data asli (x) dengan nilai minimum pada kolom tersebut (x_{min}), kemudian hasilnya dibagi dengan selisih antara nilai maksimum (x_{max}) dan nilai minimumnya ($x_{max} - x_{min}$). Sebagai ilustrasi pada variabel DBD, jika diketahui nilai $x = 52$ dengan $x_{min} = 1$ dan $x_{max} = 112$, maka perhitungan sebagai berikut.

$$x_{norm} = \frac{52 - 1}{112 - 1}$$

$$x_{norm} = \frac{51}{111}$$

$$x_{norm} \approx 0.459459 \dots$$

Adapun hasil normalisasi data *training* selengkapnya disajikan pada Tabel 4.2 dan data *testing* pada Tabel 4.3.

Tabel 4.2 Statistika Deskriptif Hasil Normalisasi Data *Training*

Variabel	Mean	Median	Varians	Min	Max
Kasus DBD	0.280	0.126	0.085	0.00	1.00
Curah Hujan (mm)	0.204	0.175	0.049	0.00	1.00
Suhu Udara (°C)	0.546	0.565	0.045	0.00	1.00
Kelembapan Udara (%)	0.669	0.684	0.038	0.00	1.00

Berdasarkan Tabel 4.2, statistika deskriptif yang disajikan merupakan hasil dari proses normalisasi pada data *training* menggunakan metode *Min-Max Scaling*. Seluruh variabel memiliki nilai minimum 0 dan maksimum 1, yang menunjukkan bahwa proses normalisasi telah diterapkan dengan baik pada data latih. Dengan kondisi tersebut, data *training* berada pada skala yang seragam sehingga siap digunakan pada tahap pemodelan selanjutnya.

Tabel 4.3 Statistika Deskriptif Hasil Normalisasi Data *Testing*

Variabel	Mean	Median	Varians	Min	Max
Kasus DBD	0.556	0.229	0.251	0.00	1.00
Curah Hujan (mm)	0.237	0.198	0.034	0.00	1.00
Suhu Udara (°C)	0.777	0.717	0.060	0.00	1.00
Kelembapan Udara (%)	0.605	0.633	0.041	0.00	1.00

Berdasarkan Tabel 4.3, statistika deskriptif yang disajikan merupakan hasil dari proses normalisasi pada data *testing*. Nilai minimum dan maksimum pada data *testing* seluruhnya berada pada rentang 0 hingga 1, yang menunjukkan bahwa data normalisasi menggunakan *Min-Max Scaling*. Hal ini menegaskan bahwa normalisasi telah dilakukan dengan benar karena sudah berada pada rentang 0 hingga 1 pada setiap variabelnya.

4.2.2 Reshaping Data

Pada tahap ini dilakukan proses *reshaping* data agar sesuai dengan format input model GRU yaitu [*samples, timesteps, features*]. Data latih dan data uji diubah menggunakan *sliding window* dengan *timesteps* = 6. Setiap input terdiri dari empat variabel fitur yaitu *lag* kasus DBD (X_1), curah hujan (X_2), suhu udara (X_3), dan kelembapan udara (X_4). Variabel X_1 berperan penting sebagai data historis target (*lagged features*) yang memungkinkan model mempelajari tren perkembangan kasus dari bulan-bulan sebelumnya untuk memprediksi nilai

periode berikutnya. Pemilihan *timesteps* = 6 didapatkan dari beberapa *timestep* yang diuji coba, sehingga dipilih hasil yang terbaik. Adapun hasil beberapa *timestep* pada Tabel 4.4 sebagai berikut.

Tabel 4.4 Uji Coba Penentuan *Timestep*

Jumlah <i>Timestep</i>	Jumlah Unit	Jumlah Lapisan	<i>Learning Rate</i>	<i>Dropout Rate</i>	<i>Batch Size</i>	<i>Epoch</i>	MAPE
3	32	1	0.001	0.2	16	100	39.69%
6	32	2	0.001	0.1	8	100	29.29%
10	32	2	0.001	0.1	8	100	39.12%
12	32	1	0.001	0.1	16	100	42.82%

Berdasarkan percobaan beberapa *timestep* dihasilkan nilai error terkecil pada *timestep* = 6. Hasil *reshaping* data menunjukkan bahwa data input latih memiliki ukuran (98, 6, 4), yang berarti terdapat 98 sampel data yang akan digunakan sebagai *training*, masing-masing menggunakan 6 periode waktu sebelumnya dengan 4 variabel input. Angka 98 diperoleh dari hasil pengurangan total 104 data latih dengan 6 *time steps*, karena diperlukan 6 data awal sebagai pemantauan untuk menghasilkan satu target prediksi pertama. Sementara itu, data target latih berukuran (98, 1) dengan 1 merupakan variabel input yang diprediksi.

Pada data uji, data input memiliki ukuran (20, 6, 4) dan data target berukuran (20, 1). Jumlah 20 sampel pada data uji ini didapat dari total 26 data uji dikurangi dengan 6 *time steps* yang digunakan dalam metode *sliding window*.

Penerapan nilai *time steps* yang konsisten antara data latih dan data uji merupakan syarat dalam arsitektur GRU. Hal ini dikarenakan model telah mengadaptasi struktur input tertentu selama proses pelatihan untuk mengenali pola keterkaitan antar waktu. Jika model dilatih untuk memahami hubungan data berdasarkan 6 periode waktu sebelumnya, maka pada tahap pengujian model

harus menerima dimensi input yang sama agar dapat menerapkan pola tersebut dengan akurat tanpa mengalami ketidaksesuaian dimensi.

Tabel 4.5 *Reshaping Data GRU*

Jenis Data	Ukuran Data	Keterangan
Data Input Latih	(98, 6, 4)	Data latih berupa variabel input yang disusun dalam 6 <i>time steps</i> dengan 4 variabel (kasus DBD, curah hujan, suhu udara, kelembapan udara)
Data Target Latih	(98, 1)	Data latih berupa nilai target yang akan diprediksi
Data Input Uji	(20, 6, 4)	Data uji berupa variabel input yang disusun dalam 6 <i>time steps</i> dengan 4 variabel (kasus DBD, curah hujan, suhu udara, kelembapan udara)
Data Target Uji	(20, 1)	Data uji berupa nilai target yang akan diprediksi

4.3 Inisialisasi *Hyperparameter*

Hyperparameter merupakan salah satu faktor yang berperan penting dalam menentukan performa model GRU pada saat proses pelatihan. Pengaturan *hyperparameter* yang tepat dapat membantu model belajar secara lebih optimal dalam mengenali pola pada data yang digunakan. Setelah dilakukan pencarian menggunakan metode *grid search*, didapatkan kombinasi parameter terbaik sebagai berikut yang disajikan dalam Tabel 4.6. Metode *grid search* dilakukan dengan cara menentukan daftar nilai kandidat untuk setiap *hyperparameter*, kemudian sistem melakukan iterasi otomatis untuk menguji setiap kombinasi nilai tersebut satu per satu pada model. Melalui proses ini, performa dari setiap kombinasi dievaluasi berdasarkan nilai *error* terkecil, sehingga diperoleh konfigurasi yang paling optimal untuk digunakan dalam arsitektur GRU guna memprediksi kasus DBD.

Tabel 4.6 Inisialisasi *Hyperparameter* GRU

Inisialisasi <i>Hyperparameter</i>	Nilai
Jumlah Unit	64
Jumlah Lapisan	2
<i>Learning Rate</i>	1×10^{-3}
<i>Dropout Rate</i>	0.2
<i>Batch Size</i>	8
<i>Epoch</i>	100
β_1	0.9
β_2	0.999
ε	1×10^{-7}

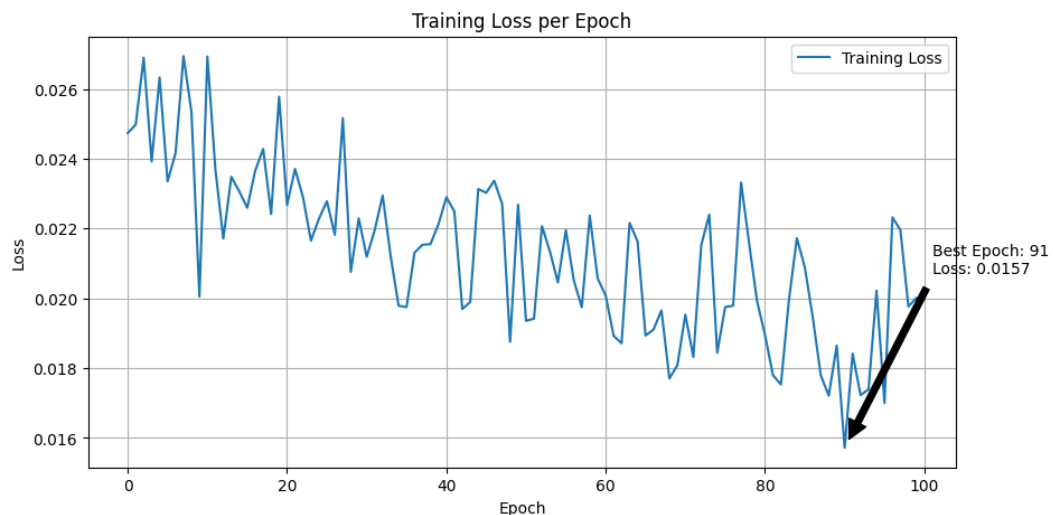
4.4 Proses *Training* GRU

Pengujian dilakukan untuk menentukan kinerja algoritma pelatihan dengan membandingkan tingkat *error* yang dihasilkan. Nilai *error* yang diperoleh dari proses pengujian disajikan pada Tabel 4.7. Pada penelitian ini, ukuran kesalahan yang digunakan adalah *Mean Square Error* (MSE) yang berfungsi sebagai *stopping criteria* selama proses pelatihan.

Tabel 4.7 Nilai Iterasi *Epoch* dan *Loss*

<i>Epoch</i>	<i>Loss</i>
1	0.0247
5	0.0263
10	0.0200
15	0.0231
20	0.0258
25	0.0223
35	0.0198
50	0.0227
60	0.0208
65	0.0216
70	0.0181
75	0.0184
80	0.0199
90	0.0186
91	0.0157
100	0.0200

Nilai *loss* terendah diperoleh pada *epoch* ke-91, yaitu sebesar 0.0157, yang menunjukkan bahwa rata-rata kesalahan antara nilai prediksi dan nilai aktual relatif kecil. Hal ini juga disajikan secara visual pada plot grafik *loss* di Gambar 4.5. Dengan demikian, nilai MSE sebesar 0.0157 menunjukkan bahwa model mampu mempelajari pola data dengan baik tanpa mengalami *overfitting*.



Gambar 4.5 Plot Iterasi *Epoch* dan *Loss*

Berdasarkan Gambar 4.5, nilai *training loss* menunjukkan pola penurunan yang tidak sepenuhnya mulus karena terdapat fluktuasi pada beberapa *epoch*. Kondisi ini terjadi karena proses pembaruan bobot menggunakan *optimizer* Adam yang melakukan penyesuaian *learning rate* secara adaptif pada setiap parameter. Akibatnya, proses optimasi tidak selalu bergerak langsung menuju titik minimum, melainkan dapat mengalami kenaikan dan penurunan kecil selama pencarian bobot optimal. Selain itu, karakteristik data deret waktu yang memiliki jumlah data pelatihan terbatas menyebabkan variasi gradien pada setiap iterasi sehingga menghasilkan fluktuasi nilai *loss*. Meskipun demikian, fluktuasi yang terjadi relatif kecil dan nilai *training loss* secara keseluruhan menunjukkan tren menurun hingga mencapai nilai minimum sebesar 0.0157 pada epoch ke-91.

4.5 Perhitungan Matematis Model GRU

Pada tahap ini, akan dilakukan simulasi perhitungan GRU untuk setiap *time step*. Perhitungan ini bertujuan untuk memahami bagaimana model GRU memproses data historis secara berulang melalui mekanisme *gate-gate* utama yang terdapat dalam arsitektur GRU yaitu *update gate*, *reset gate*, *candidate hidden state* dan *hidden state update*.

Data input pada *timestep* ke-*t* diperoleh dari hasil normalisasi data variabel penelitian, yaitu jumlah kasus DBD, curah hujan, suhu udara, dan kelembapan udara. Pada penelitian ini simulasi perhitungan ini yang digunakan yaitu dengan jumlah *time step* sebanyak 6 untuk prediksi. Adapun contoh perhitungan sebagai berikut.

Diketahui:

No	Curah Hujan	Suhu Udara	Kelembapan Udara	Kasus DBD
1	0.742	0.615	0.801	0.693
2	0.688	0.632	0.784	0.655
3	0.801	0.598	0.842	0.744
4	0.756	0.607	0.825	0.721
5	0.690	0.620	0.790	0.670
6	0.825	0.590	0.856	0.768
7	0.774	0.601	0.831	0.734
8	0.842	0.584	0.870	0.795
9	0.790	0.596	0.846	0.752
10	0.856	0.579	0.884	0.808

Hidden state sebelumnya diasumsikan bernilai:

$$h_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Hal ini dilakukan karena pada perhitungan *timestep* pertama belum terdapat nilai *hidden state* sebelumnya. Pada tahap awal, nilai bobot dan bias akan

ditentukan secara acak oleh program dan akan terus diperbarui secara berkala menggunakan algoritma *Optimizer* Adam dengan tujuan meminimalkan *error* dalam prediksi.

Layer 1 :

1. *Time step* ke-1 ($t = 1$)

$$x_1 = \begin{bmatrix} 0.742 \\ 0.615 \\ 0.801 \\ 0.693 \end{bmatrix}$$

$$h_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Menghitung *update gate* menggunakan Persamaan (2.2)

Diketahui:

$$W_z = \begin{bmatrix} 0.214 & 0.132 & 0.365 & 0.287 \\ 0.165 & 0.298 & 0.154 & 0.241 \\ 0.321 & 0.187 & 0.265 & 0.143 \\ 0.278 & 0.311 & 0.176 & 0.298 \end{bmatrix}$$

$$U_z = \begin{bmatrix} 0.145 & 0.087 & 0.165 & 0.092 \\ 0.101 & 0.154 & 0.118 & 0.133 \\ 0.176 & 0.144 & 0.109 & 0.021 \\ 0.134 & 0.165 & 0.142 & 0.154 \end{bmatrix}$$

$$b_z = \begin{bmatrix} 0.101 \\ 0.102 \\ 0.099 \\ 0.103 \end{bmatrix}$$

Langkah perhitungan:

$$z_1 = \sigma(W_z x_1 + U_z h_0 + b_z)$$

$$z_1 = \sigma \left(\begin{bmatrix} 0.214 & 0.132 & 0.365 & 0.287 \\ 0.165 & 0.298 & 0.154 & 0.241 \\ 0.321 & 0.187 & 0.265 & 0.143 \\ 0.278 & 0.311 & 0.176 & 0.298 \end{bmatrix} \begin{bmatrix} 0.742 \\ 0.615 \\ 0.801 \\ 0.693 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.101 \\ 0.102 \\ 0.099 \\ 0.103 \end{bmatrix} \right)$$

$$z_1 = \sigma \left(\begin{bmatrix} 0.724 \\ 0.595 \\ 0.664 \\ 0.744 \end{bmatrix} + \begin{bmatrix} 0.101 \\ 0.102 \\ 0.099 \\ 0.103 \end{bmatrix} \right)$$

$$z_1 = \sigma \begin{bmatrix} 0.825 \\ 0.697 \\ 0.763 \\ 0.847 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$z_1 = \begin{bmatrix} 0.695 \\ 0.667 \\ 0.682 \\ 0.700 \end{bmatrix}$$

Menghitung *reset gate* menggunakan Persamaan (2.3)

Diketahui:

$$W_r = \begin{bmatrix} 0.243 & 0.176 & 0.298 & 0.254 \\ 0.198 & 0.267 & 0.187 & 0.211 \\ 0.287 & 0.154 & 0.232 & 0.176 \\ 0.256 & 0.289 & 0.198 & 0.265 \end{bmatrix}$$

$$U_r = \begin{bmatrix} 0.154 & 0.098 & 0.143 & 0.087 \\ 0.132 & 0.143 & 0.101 & 0.121 \\ 0.165 & 0.132 & 0.118 & 0.109 \\ 0.143 & 0.154 & 0.132 & 0.143 \end{bmatrix}$$

$$b_r = \begin{bmatrix} 0.111 \\ 0.105 \\ 0.107 \\ 0.109 \end{bmatrix}$$

Langkah perhitungan:

$$r_1 = \sigma(W_r x_1 + U_r h_0 + b_r)$$

$$r_1 = \sigma \left(\begin{bmatrix} 0.243 & 0.176 & 0.298 & 0.254 \\ 0.198 & 0.267 & 0.187 & 0.211 \\ 0.287 & 0.154 & 0.232 & 0.176 \\ 0.256 & 0.289 & 0.198 & 0.265 \end{bmatrix} \begin{bmatrix} 0.742 \\ 0.615 \\ 0.801 \\ 0.693 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.111 \\ 0.105 \\ 0.107 \\ 0.109 \end{bmatrix} \right)$$

$$r_1 = \sigma \left(\begin{bmatrix} 0.749 \\ 0.611 \\ 0.635 \\ 0.726 \end{bmatrix} + \begin{bmatrix} 0.111 \\ 0.105 \\ 0.107 \\ 0.109 \end{bmatrix} \right)$$

$$r_1 = \sigma \begin{bmatrix} 0.860 \\ 0.716 \\ 0.742 \\ 0.835 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$r_1 = \begin{bmatrix} 0.703 \\ 0.672 \\ 0.677 \\ 0.697 \end{bmatrix}$$

Menghitung *candidate hidden state* menggunakan Persamaan (2.4)

Diketahui:

$$W_h = \begin{bmatrix} 0.298 & 0.187 & 0.276 & 0.243 \\ 0.176 & 0.298 & 0.165 & 0.232 \\ 0.265 & 0.154 & 0.243 & 0.176 \\ 0.254 & 0.276 & 0.187 & 0.243 \end{bmatrix}$$

$$U_h = \begin{bmatrix} 0.132 & 0.109 & 0.143 & 0.098 \\ 0.121 & 0.143 & 0.109 & 0.132 \\ 0.154 & 0.121 & 0.132 & 0.109 \\ 0.143 & 0.130 & 0.121 & 0.143 \end{bmatrix}$$

$$b_h = \begin{bmatrix} 0.115 \\ 0.112 \\ 0.111 \\ 0.114 \end{bmatrix}$$

$$r_1 = \begin{bmatrix} 0.703 \\ 0.672 \\ 0.677 \\ 0.697 \end{bmatrix}$$

Langkah perhitungan:

$$\tilde{h}_1 = \tanh (W_h x_1 + U_h (r_1 \odot h_0) + b_h)$$

$$\tilde{h}_1 = \tanh \left(\begin{bmatrix} 0.298 & 0.187 & 0.276 & 0.243 \\ 0.176 & 0.298 & 0.165 & 0.232 \\ 0.265 & 0.154 & 0.243 & 0.176 \\ 0.254 & 0.276 & 0.187 & 0.243 \end{bmatrix} \begin{bmatrix} 0.742 \\ 0.615 \\ 0.801 \\ 0.693 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.115 \\ 0.112 \\ 0.111 \\ 0.114 \end{bmatrix} \right)$$

$$\tilde{h}_1 = \tanh \left(\begin{bmatrix} 0.325 \\ 0.705 \\ 0.285 \\ 0.175 \end{bmatrix} + \begin{bmatrix} 0.115 \\ 0.112 \\ 0.111 \\ 0.114 \end{bmatrix} \right)$$

$$\tilde{h}_1 = \tanh \begin{bmatrix} 0.440 \\ 0.817 \\ 0.639 \\ 0.692 \end{bmatrix}$$

Fungsi tanh didefinisikan sebagai:

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

Hasil:

$$\tilde{h}_1 = \begin{bmatrix} 0.702 \\ 0.624 \\ 0.639 \\ 0.692 \end{bmatrix}$$

Menghitung *hidden state update* menggunakan Persamaan (2.5)

Diketahui:

$$z_1 = \begin{bmatrix} 0.695 \\ 0.667 \\ 0.682 \\ 0.700 \end{bmatrix}$$

$$\tilde{h}_1 = \begin{bmatrix} 0.702 \\ 0.624 \\ 0.639 \\ 0.692 \end{bmatrix}$$

Substitusi nilai ke dalam persamaan menghasilkan:

$$h_1 = (1 - z_1) \odot h_0 + z_1 \odot \tilde{h}_1$$

$$h_1 = \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.695 \\ 0.667 \\ 0.682 \\ 0.700 \end{bmatrix} \right) \odot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.695 \\ 0.667 \\ 0.682 \\ 0.700 \end{bmatrix} \odot \begin{bmatrix} 0.702 \\ 0.624 \\ 0.639 \\ 0.692 \end{bmatrix}$$

$$h_1 = \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix}$$

2. *Time step* ke-2 ($t = 2$)

$$x_2 = \begin{bmatrix} 0.688 \\ 0.632 \\ 0.784 \\ 0.655 \end{bmatrix}; y_2 = 0.744$$

$$h_1 = \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix}$$

Menghitung *update gate* menggunakan Persamaan (2.2)

Diketahui:

$$W_z = \begin{bmatrix} 0.214 & 0.132 & 0.365 & 0.287 \\ 0.165 & 0.298 & 0.154 & 0.241 \\ 0.321 & 0.187 & 0.265 & 0.143 \\ 0.278 & 0.311 & 0.176 & 0.298 \end{bmatrix}$$

$$U_z = \begin{bmatrix} 0.145 & 0.087 & 0.165 & 0.092 \\ 0.101 & 0.154 & 0.118 & 0.133 \\ 0.176 & 0.144 & 0.109 & 0.021 \\ 0.134 & 0.165 & 0.142 & 0.154 \end{bmatrix}$$

$$b_z = \begin{bmatrix} 0.101 \\ 0.102 \\ 0.099 \\ 0.103 \end{bmatrix}$$

Langkah perhitungan:

$$z_2 = \sigma(W_z x_2 + U_z h_1 + b_z)$$

$$z_2 = \sigma \left(\begin{bmatrix} 0.214 & 0.132 & 0.365 & 0.287 \\ 0.165 & 0.298 & 0.154 & 0.241 \\ 0.321 & 0.187 & 0.265 & 0.143 \\ 0.278 & 0.311 & 0.176 & 0.298 \end{bmatrix} \begin{bmatrix} 0.688 \\ 0.632 \\ 0.784 \\ 0.655 \end{bmatrix} + \begin{bmatrix} 0.145 & 0.087 & 0.165 & 0.092 \\ 0.101 & 0.154 & 0.118 & 0.133 \\ 0.176 & 0.144 & 0.109 & 0.021 \\ 0.134 & 0.165 & 0.142 & 0.154 \end{bmatrix} \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix} + \begin{bmatrix} 0.101 \\ 0.102 \\ 0.099 \\ 0.103 \end{bmatrix} \right)$$

$$z_2 = \sigma \left(\begin{bmatrix} 0.697 \\ 0.581 \\ 0.641 \\ 0.721 \end{bmatrix} + \begin{bmatrix} 0.224 \\ 0.228 \\ 0.253 \\ 0.271 \end{bmatrix} + \begin{bmatrix} 0.101 \\ 0.102 \\ 0.099 \\ 0.103 \end{bmatrix} \right)$$

$$z_2 = \sigma \begin{bmatrix} 1.022 \\ 0.911 \\ 0.993 \\ 1.095 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$z_2 = \begin{bmatrix} 0.409 \\ 0.365 \\ 0.350 \\ 0.335 \end{bmatrix}$$

Menghitung *reset gate* menggunakan Persamaan (2.3)

Diketahui:

$$W_r = \begin{bmatrix} 0.243 & 0.176 & 0.298 & 0.254 \\ 0.198 & 0.267 & 0.187 & 0.211 \\ 0.287 & 0.154 & 0.232 & 0.176 \\ 0.256 & 0.289 & 0.198 & 0.265 \end{bmatrix}$$

$$U_r = \begin{bmatrix} 0.154 & 0.098 & 0.143 & 0.087 \\ 0.132 & 0.143 & 0.101 & 0.121 \\ 0.165 & 0.132 & 0.118 & 0.109 \\ 0.143 & 0.154 & 0.132 & 0.143 \end{bmatrix}$$

$$b_r = \begin{bmatrix} 0.111 \\ 0.105 \\ 0.107 \\ 0.109 \end{bmatrix}$$

Langkah perhitungan:

$$r_2 = \sigma(W_r x_2 + U_r h_1 + b_r)$$

$$r_2 = \sigma \left(\begin{bmatrix} 0.243 & 0.176 & 0.298 & 0.254 \\ 0.198 & 0.267 & 0.187 & 0.211 \\ 0.287 & 0.154 & 0.232 & 0.176 \\ 0.256 & 0.289 & 0.198 & 0.265 \end{bmatrix} \begin{bmatrix} 0.688 \\ 0.632 \\ 0.784 \\ 0.655 \end{bmatrix} + \begin{bmatrix} 0.154 & 0.098 & 0.143 & 0.087 \\ 0.132 & 0.143 & 0.101 & 0.121 \\ 0.165 & 0.132 & 0.118 & 0.109 \\ 0.143 & 0.154 & 0.132 & 0.143 \end{bmatrix} \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix} + \begin{bmatrix} 0.111 \\ 0.105 \\ 0.107 \\ 0.109 \end{bmatrix} \right)$$

$$r_2 = \sigma \left(\begin{bmatrix} 0.741 \\ 0.618 \\ 0.635 \\ 0.718 \end{bmatrix} + \begin{bmatrix} 0.216 \\ 0.218 \\ 0.231 \\ 0.272 \end{bmatrix} + \begin{bmatrix} 0.111 \\ 0.105 \\ 0.107 \\ 0.109 \end{bmatrix} \right)$$

$$r_2 = \sigma \begin{bmatrix} 1.068 \\ 0.941 \\ 0.973 \\ 1.099 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$r_2 = \begin{bmatrix} 0.744 \\ 0.719 \\ 0.726 \\ 0.750 \end{bmatrix}$$

Menghitung *candidate hidden state* menggunakan Persamaan (2.4)

Diketahui:

$$W_h = \begin{bmatrix} 0.298 & 0.187 & 0.276 & 0.243 \\ 0.176 & 0.298 & 0.165 & 0.232 \\ 0.265 & 0.154 & 0.243 & 0.176 \\ 0.254 & 0.276 & 0.187 & 0.243 \end{bmatrix}$$

$$U_h = \begin{bmatrix} 0.132 & 0.109 & 0.143 & 0.098 \\ 0.121 & 0.143 & 0.109 & 0.132 \\ 0.154 & 0.121 & 0.132 & 0.109 \\ 0.143 & 0.130 & 0.121 & 0.143 \end{bmatrix}$$

$$b_h = \begin{bmatrix} 0.115 \\ 0.112 \\ 0.111 \\ 0.114 \end{bmatrix}$$

$$r_2 = \begin{bmatrix} 0.744 \\ 0.719 \\ 0.726 \\ 0.750 \end{bmatrix}$$

Langkah perhitungan:

$$\tilde{h}_2 = \tanh (W_h x_2 + U_h(r_2 \odot h_1) + b_h)$$

$$\begin{aligned} \tilde{h}_2 = \tanh & \left(\begin{bmatrix} 0.298 & 0.187 & 0.276 & 0.243 \\ 0.176 & 0.298 & 0.165 & 0.232 \\ 0.265 & 0.154 & 0.243 & 0.176 \\ 0.254 & 0.276 & 0.187 & 0.243 \end{bmatrix} \begin{bmatrix} 0.688 \\ 0.632 \\ 0.784 \\ 0.655 \end{bmatrix} \right. \\ & + \begin{bmatrix} 0.132 & 0.109 & 0.143 & 0.098 \\ 0.121 & 0.143 & 0.109 & 0.132 \\ 0.154 & 0.121 & 0.132 & 0.109 \\ 0.143 & 0.130 & 0.121 & 0.143 \end{bmatrix} \left(\begin{bmatrix} 0.744 \\ 0.719 \\ 0.726 \\ 0.750 \end{bmatrix} \odot \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix} \right) \\ & \left. + \begin{bmatrix} 0.115 \\ 0.112 \\ 0.111 \\ 0.114 \end{bmatrix} \right) \end{aligned}$$

$$\tilde{h}_2 = \tanh \left(\begin{bmatrix} 0.756 \\ 0.628 \\ 0.648 \\ 0.736 \end{bmatrix} + \begin{bmatrix} 0.154 \\ 0.159 \\ 0.168 \\ 0.179 \end{bmatrix} + \begin{bmatrix} 0.115 \\ 0.112 \\ 0.111 \\ 0.114 \end{bmatrix} \right)$$

$$\tilde{h}_2 = \tanh \begin{bmatrix} 1.025 \\ 0.899 \\ 0.927 \\ 1.029 \end{bmatrix}$$

Fungsi tanh didefinisikan sebagai:

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

Hasil:

$$\tilde{h}_2 = \begin{bmatrix} 0.772 \\ 0.716 \\ 0.729 \\ 0.773 \end{bmatrix}$$

Menghitung *hidden state update* menggunakan Persamaan (2.5)

Diketahui:

$$z_2 = \begin{bmatrix} 0.409 \\ 0.365 \\ 0.350 \\ 0.335 \end{bmatrix}$$

$$\tilde{h}_2 = \begin{bmatrix} 0.772 \\ 0.716 \\ 0.729 \\ 0.773 \end{bmatrix}$$

Substitusi nilai ke dalam persamaan menghasilkan:

$$h_2 = (1 - z_2) \odot h_1 + z_2 \odot \tilde{h}_2$$

$$h_2 = \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.409 \\ 0.365 \\ 0.350 \\ 0.335 \end{bmatrix} \right) \odot \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix} + \begin{bmatrix} 0.409 \\ 0.365 \\ 0.350 \\ 0.335 \end{bmatrix} \odot \begin{bmatrix} 0.772 \\ 0.716 \\ 0.729 \\ 0.773 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 0.129 \\ 0.119 \\ 0.118 \\ 0.121 \end{bmatrix} + \begin{bmatrix} 0.567 \\ 0.511 \\ 0.532 \\ 0.579 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 0.696 \\ 0.630 \\ 0.650 \\ 0.700 \end{bmatrix}$$

Dengan perhitungan yang sama, didapatkan hasil:

<i>Hidden State</i>	Nilai
h_1	[0.488 0.416 0.436 0.484]
h_2	[0.696 0.630 0.650 0.700]
h_3	[0.648 0.560 0.578 0.636]
h_4	[0.705 0.624 0.641 0.704]
h_5	[0.734 0.661 0.676 0.739]
h_6	[0.749 0.681 0.694 0.756]

Layer 2 :

Output *hidden state* dari *layer* pertama akan menjadi input untuk *layer* kedua.

1. *Time step* ke-1 ($t = 1$)

$$x_1^{(2)} = h_1 = \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix}$$

$$h_0^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Menghitung *update gate* menggunakan Persamaan (2.2)

Diketahui:

$$W_z^{(2)} = \begin{bmatrix} 0.254 & 0.176 & 0.298 & 0.243 \\ 0.198 & 0.287 & 0.165 & 0.232 \\ 0.276 & 0.154 & 0.243 & 0.176 \\ 0.243 & 0.256 & 0.187 & 0.254 \end{bmatrix}$$

$$U_z^{(2)} = \begin{bmatrix} 0.132 & 0.109 & 0.143 & 0.098 \\ 0.121 & 0.143 & 0.109 & 0.132 \\ 0.154 & 0.121 & 0.132 & 0.109 \\ 0.143 & 0.132 & 0.121 & 0.143 \end{bmatrix}$$

$$b_z^{(2)} = \begin{bmatrix} 0.104 \\ 0.103 \\ 0.101 \\ 0.105 \end{bmatrix}$$

Langkah perhitungan:

$$z_1^{(2)} = \sigma(W_z^{(2)}x_1 + U_z^{(2)}h_0 + b_z^{(2)})$$

$$z_1^{(2)} = \sigma \left(\begin{bmatrix} 0.254 & 0.176 & 0.298 & 0.243 \\ 0.198 & 0.287 & 0.165 & 0.232 \\ 0.276 & 0.154 & 0.243 & 0.176 \\ 0.243 & 0.256 & 0.187 & 0.254 \end{bmatrix} \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.104 \\ 0.103 \\ 0.101 \\ 0.105 \end{bmatrix} \right)$$

$$z_1^{(2)} = \sigma \left(\begin{bmatrix} 0.445 \\ 0.400 \\ 0.390 \\ 0.539 \end{bmatrix} + \begin{bmatrix} 0.104 \\ 0.103 \\ 0.101 \\ 0.105 \end{bmatrix} \right)$$

$$z_1^{(2)} = \sigma \begin{bmatrix} 0.549 \\ 0.503 \\ 0.491 \\ 0.539 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$z_1^{(2)} = \begin{bmatrix} 0.634 \\ 0.623 \\ 0.620 \\ 0.632 \end{bmatrix}$$

Menghitung *reset gate* menggunakan Persamaan (2.3)

Diketahui:

$$W_r^{(2)} = \begin{bmatrix} 0.243 & 0.198 & 0.287 & 0.254 \\ 0.176 & 0.265 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.232 & 0.276 & 0.198 & 0.243 \end{bmatrix}$$

$$U_r^{(2)} = \begin{bmatrix} 0.121 & 0.132 & 0.109 & 0.121 \\ 0.109 & 0.143 & 0.132 & 0.109 \\ 0.143 & 0.121 & 0.154 & 0.132 \\ 0.132 & 0.154 & 0.121 & 0.143 \end{bmatrix}$$

$$b_r^{(2)} = \begin{bmatrix} 0.108 \\ 0.107 \\ 0.106 \\ 0.109 \end{bmatrix}$$

Langkah perhitungan:

$$r_1^{(2)} = \sigma(W_r^{(2)} x_1 + U_r^{(2)} h_0 + b_r^{(2)})$$

$$r_1^{(2)} = \sigma \left(\begin{bmatrix} 0.243 & 0.198 & 0.287 & 0.254 \\ 0.176 & 0.265 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.232 & 0.276 & 0.198 & 0.243 \end{bmatrix} \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.108 \\ 0.107 \\ 0.106 \\ 0.109 \end{bmatrix} \right)$$

$$r_1^{(2)} = \sigma \left(\begin{bmatrix} 0.449 \\ 0.404 \\ 0.397 \\ 0.439 \end{bmatrix} + \begin{bmatrix} 0.108 \\ 0.107 \\ 0.106 \\ 0.109 \end{bmatrix} \right)$$

$$r_1^{(2)} = \sigma \begin{bmatrix} 0.557 \\ 0.511 \\ 0.503 \\ 0.548 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$r_1^{(2)} = \begin{bmatrix} 0.636 \\ 0.625 \\ 0.623 \\ 0.634 \end{bmatrix}$$

Menghitung *candidate hidden state* menggunakan Persamaan (2.4)

Diketahui:

$$W_h^{(2)} = \begin{bmatrix} 0.287 & 0.176 & 0.265 & 0.243 \\ 0.198 & 0.276 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.243 & 0.265 & 0.187 & 0.254 \end{bmatrix}$$

$$U_h^{(2)} = \begin{bmatrix} 0.132 & 0.121 & 0.109 & 0.132 \\ 0.109 & 0.143 & 0.121 & 0.109 \\ 0.143 & 0.132 & 0.154 & 0.121 \\ 0.121 & 0.143 & 0.132 & 0.143 \end{bmatrix}$$

$$b_h^{(2)} = \begin{bmatrix} 0.112 \\ 0.111 \\ 0.113 \\ 0.114 \end{bmatrix}$$

$$r_1^{(2)} = \begin{bmatrix} 0.636 \\ 0.625 \\ 0.623 \\ 0.634 \end{bmatrix}$$

Langkah perhitungan:

$$\tilde{h}_1^{(2)} = \tanh (W_h^{(2)} x_1 + U_h^{(2)} (r_1^{(2)} \odot h_0) + b_h^{(2)})$$

$$\tilde{h}_1^{(2)} = \tanh \left(\begin{bmatrix} 0.287 & 0.176 & 0.265 & 0.243 \\ 0.198 & 0.276 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.243 & 0.265 & 0.187 & 0.254 \end{bmatrix} \begin{bmatrix} 0.488 \\ 0.416 \\ 0.436 \\ 0.484 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.112 \\ 0.111 \\ 0.113 \\ 0.114 \end{bmatrix} \right)$$

$$\tilde{h}_1^{(2)} = \tanh \left(\begin{bmatrix} 0.449 \\ 0.404 \\ 0.394 \\ 0.438 \end{bmatrix} + \begin{bmatrix} 0.112 \\ 0.111 \\ 0.113 \\ 0.114 \end{bmatrix} \right)$$

$$\tilde{h}_1^{(2)} = \tanh \begin{bmatrix} 0.561 \\ 0.515 \\ 0.507 \\ 0.552 \end{bmatrix}$$

Fungsi tanh didefinisikan sebagai:

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

Hasil:

$$\tilde{h}_1^{(2)} = \begin{bmatrix} 0.509 \\ 0.474 \\ 0.467 \\ 0.502 \end{bmatrix}$$

Menghitung *hidden state update* menggunakan Persamaan (2.5)

Diketahui:

$$z_1^{(2)} = \begin{bmatrix} 0.634 \\ 0.623 \\ 0.620 \\ 0.632 \end{bmatrix}$$

$$\tilde{h}_1^{(2)} = \begin{bmatrix} 0.509 \\ 0.474 \\ 0.467 \\ 0.502 \end{bmatrix}$$

Substitusi nilai ke dalam persamaan menghasilkan:

$$h_1^{(2)} = (1 - z_1^{(2)}) \odot h_0 + z_1 \odot \tilde{h}_1$$

$$h_1^{(2)} = \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.634 \\ 0.623 \\ 0.620 \\ 0.632 \end{bmatrix} \right) \odot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.634 \\ 0.623 \\ 0.620 \\ 0.632 \end{bmatrix} \odot \begin{bmatrix} 0.509 \\ 0.474 \\ 0.467 \\ 0.502 \end{bmatrix}$$

$$h_1^{(2)} = \begin{bmatrix} 0.323 \\ 0.295 \\ 0.290 \\ 0.317 \end{bmatrix}$$

2. *Time step* ke-2 ($t = 2$)

$$x_2^{(2)} = \begin{bmatrix} 0.696 \\ 0.630 \\ 0.650 \\ 0.700 \end{bmatrix}$$

$$h_1^{(2)} = \begin{bmatrix} 0.323 \\ 0.295 \\ 0.290 \\ 0.317 \end{bmatrix}$$

Menghitung *update gate* menggunakan Persamaan (2.2)

Diketahui:

$$W_z^{(2)} = \begin{bmatrix} 0.254 & 0.176 & 0.298 & 0.243 \\ 0.198 & 0.287 & 0.165 & 0.232 \\ 0.276 & 0.154 & 0.243 & 0.176 \\ 0.243 & 0.256 & 0.187 & 0.254 \end{bmatrix}$$

$$U_z^{(2)} = \begin{bmatrix} 0.132 & 0.109 & 0.143 & 0.098 \\ 0.121 & 0.143 & 0.109 & 0.132 \\ 0.154 & 0.121 & 0.132 & 0.109 \\ 0.143 & 0.132 & 0.121 & 0.143 \end{bmatrix}$$

$$b_z^{(2)} = \begin{bmatrix} 0.104 \\ 0.103 \\ 0.101 \\ 0.105 \end{bmatrix}$$

Langkah perhitungan:

$$z_2^{(2)} = \sigma(W_z x_2 + U_z h_1 + b_z)$$

$$z_2^{(2)} = \sigma \left(\begin{bmatrix} 0.254 & 0.176 & 0.298 & 0.243 \\ 0.198 & 0.287 & 0.165 & 0.232 \\ 0.276 & 0.154 & 0.243 & 0.176 \\ 0.243 & 0.256 & 0.187 & 0.254 \end{bmatrix} \begin{bmatrix} 0.696 \\ 0.630 \\ 0.650 \\ 0.700 \end{bmatrix} + \begin{bmatrix} 0.132 & 0.109 & 0.143 & 0.098 \\ 0.121 & 0.143 & 0.109 & 0.132 \\ 0.154 & 0.121 & 0.132 & 0.109 \\ 0.143 & 0.132 & 0.121 & 0.143 \end{bmatrix} \begin{bmatrix} 0.323 \\ 0.295 \\ 0.290 \\ 0.317 \end{bmatrix} + \begin{bmatrix} 0.104 \\ 0.103 \\ 0.101 \\ 0.105 \end{bmatrix} \right)$$

$$z_2^{(2)} = \sigma \left(\begin{bmatrix} 0.652 \\ 0.588 \\ 0.570 \\ 0.636 \end{bmatrix} + \begin{bmatrix} 0.147 \\ 0.155 \\ 0.159 \\ 0.165 \end{bmatrix} + \begin{bmatrix} 0.104 \\ 0.103 \\ 0.101 \\ 0.105 \end{bmatrix} \right)$$

$$z_2^{(2)} = \sigma \begin{bmatrix} 0.903 \\ 0.846 \\ 0.830 \\ 0.906 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$z_2^{(2)} = \begin{bmatrix} 0.712 \\ 0.700 \\ 0.696 \\ 0.712 \end{bmatrix}$$

Menghitung *reset gate* menggunakan Persamaan (2.3)

Diketahui:

$$W_r^{(2)} = \begin{bmatrix} 0.243 & 0.198 & 0.287 & 0.254 \\ 0.176 & 0.265 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.232 & 0.276 & 0.198 & 0.243 \end{bmatrix}$$

$$U_r^{(2)} = \begin{bmatrix} 0.121 & 0.132 & 0.109 & 0.121 \\ 0.109 & 0.143 & 0.132 & 0.109 \\ 0.143 & 0.121 & 0.154 & 0.132 \\ 0.132 & 0.154 & 0.121 & 0.143 \end{bmatrix}$$

$$b_r^{(2)} = \begin{bmatrix} 0.108 \\ 0.107 \\ 0.106 \\ 0.109 \end{bmatrix}$$

Langkah perhitungan:

$$r_2^{(2)} = \sigma(W_r^{(2)} x_2 + U_r^{(2)} h_1 + b_r^{(2)})$$

$$r_2^{(2)} = \sigma \left(\begin{bmatrix} 0.243 & 0.198 & 0.287 & 0.254 \\ 0.176 & 0.265 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.232 & 0.276 & 0.198 & 0.243 \end{bmatrix} \begin{bmatrix} 0.696 \\ 0.630 \\ 0.650 \\ 0.700 \end{bmatrix} + \begin{bmatrix} 0.121 & 0.132 & 0.109 & 0.121 \\ 0.109 & 0.143 & 0.132 & 0.109 \\ 0.143 & 0.121 & 0.154 & 0.132 \\ 0.132 & 0.154 & 0.121 & 0.143 \end{bmatrix} \begin{bmatrix} 0.323 \\ 0.295 \\ 0.290 \\ 0.317 \end{bmatrix} + \begin{bmatrix} 0.108 \\ 0.107 \\ 0.106 \\ 0.109 \end{bmatrix} \right)$$

$$r_2^{(2)} = \sigma \left(\begin{bmatrix} 0.648 \\ 0.582 \\ 0.565 \\ 0.629 \end{bmatrix} + \begin{bmatrix} 0.144 \\ 0.149 \\ 0.158 \\ 0.164 \end{bmatrix} + \begin{bmatrix} 0.108 \\ 0.107 \\ 0.106 \\ 0.109 \end{bmatrix} \right)$$

$$r_2^{(2)} = \sigma \begin{bmatrix} 0.900 \\ 0.838 \\ 0.829 \\ 0.902 \end{bmatrix}$$

Fungsi sigmoid didefinisikan sebagai:

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

Hasil:

$$r_2^{(2)} = \begin{bmatrix} 0.711 \\ 0.698 \\ 0.696 \\ 0.711 \end{bmatrix}$$

Menghitung *candidate hidden state* menggunakan Persamaan (2.4)

Diketahui:

$$W_h^{(2)} = \begin{bmatrix} 0.287 & 0.176 & 0.265 & 0.243 \\ 0.198 & 0.276 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.243 & 0.265 & 0.187 & 0.254 \end{bmatrix}$$

$$U_h^{(2)} = \begin{bmatrix} 0.132 & 0.121 & 0.109 & 0.132 \\ 0.109 & 0.143 & 0.121 & 0.109 \\ 0.143 & 0.132 & 0.154 & 0.121 \\ 0.121 & 0.143 & 0.132 & 0.143 \end{bmatrix}$$

$$b_h^{(2)} = \begin{bmatrix} 0.112 \\ 0.111 \\ 0.113 \\ 0.114 \end{bmatrix}$$

$$r_2^{(2)} = \begin{bmatrix} 0.711 \\ 0.698 \\ 0.696 \\ 0.711 \end{bmatrix}$$

Langkah perhitungan:

$$\tilde{h}_2^{(2)} = \tanh (W_h^{(2)} x_2 + U_h^{(2)} (r_2^{(2)} \odot h_1) + b_h^{(2)})$$

$$\begin{aligned} \tilde{h}_2^{(2)} = \tanh & \left(\begin{bmatrix} 0.287 & 0.176 & 0.265 & 0.243 \\ 0.198 & 0.276 & 0.176 & 0.232 \\ 0.254 & 0.165 & 0.243 & 0.187 \\ 0.243 & 0.265 & 0.187 & 0.254 \end{bmatrix} \begin{bmatrix} 0.696 \\ 0.630 \\ 0.650 \\ 0.700 \end{bmatrix} \right. \\ & + \begin{bmatrix} 0.132 & 0.121 & 0.109 & 0.132 \\ 0.109 & 0.143 & 0.121 & 0.109 \\ 0.143 & 0.132 & 0.154 & 0.121 \\ 0.121 & 0.143 & 0.132 & 0.143 \end{bmatrix} \left(\begin{bmatrix} 0.711 \\ 0.698 \\ 0.696 \\ 0.711 \end{bmatrix} \odot \begin{bmatrix} 0.323 \\ 0.295 \\ 0.290 \\ 0.317 \end{bmatrix} \right) \\ & \left. + \begin{bmatrix} 0.112 \\ 0.111 \\ 0.113 \\ 0.114 \end{bmatrix} \right) \end{aligned}$$

$$\tilde{h}_2^{(2)} = \tanh \left(\begin{bmatrix} 0.626 \\ 0.590 \\ 0.567 \\ 0.637 \end{bmatrix} + \begin{bmatrix} 0.112 \\ 0.102 \\ 0.117 \\ 0.117 \end{bmatrix} + \begin{bmatrix} 0.112 \\ 0.111 \\ 0.113 \\ 0.114 \end{bmatrix} \right)$$

$$\tilde{h}_2^{(2)} = \begin{bmatrix} 0.850 \\ 0.803 \\ 0.797 \\ 0.868 \end{bmatrix}$$

Fungsi tanh didefinisikan sebagai:

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

Hasil:

$$\tilde{h}_2^{(2)} = \begin{bmatrix} 0.691 \\ 0.666 \\ 0.662 \\ 0.700 \end{bmatrix}$$

Menghitung *hidden state update* menggunakan Persamaan (2.5)

Diketahui:

$$z_2^{(2)} = \begin{bmatrix} 0.712 \\ 0.700 \\ 0.696 \\ 0.712 \end{bmatrix}$$

$$\tilde{h}_2^{(2)} = \begin{bmatrix} 0.691 \\ 0.666 \\ 0.662 \\ 0.700 \end{bmatrix}$$

Substitusi nilai ke dalam persamaan menghasilkan:

$$h_2^{(2)} = (1 - z_2^{(2)}) \odot h_1 + z_2 \odot \tilde{h}_2^{(2)}$$

$$h_2^{(2)} = \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.712 \\ 0.700 \\ 0.696 \\ 0.712 \end{bmatrix} \right) \odot \begin{bmatrix} 0.323 \\ 0.295 \\ 0.290 \\ 0.317 \end{bmatrix} + \begin{bmatrix} 0.712 \\ 0.700 \\ 0.696 \\ 0.712 \end{bmatrix} \odot \begin{bmatrix} 0.691 \\ 0.666 \\ 0.662 \\ 0.700 \end{bmatrix}$$

$$h_2^{(2)} = \begin{bmatrix} 0.585 \\ 0.555 \\ 0.549 \\ 0.589 \end{bmatrix}$$

Dengan perhitungan yang sama, didapatkan hasil:

<i>Hidden State</i>	Nilai
$h_1^{(2)}$	([0.323 0.295 0.290 0.317])
$h_2^{(2)}$	([0.585 0.555 0.549 0.589])
$h_3^{(2)}$	([0.709 0.688 0.676 0.718])
$h_4^{(2)}$	([0.776 0.758 0.744 0.784])
$h_5^{(2)}$	([0.812 0.796 0.781 0.819])
$h_6^{(2)}$	([0.831 0.817 0.803 0.838])

Pada *layer 2*, hidden state mengalami peningkatan secara bertahap karena informasi temporal dari *layer 1* dipelajari kembali untuk memperoleh representasi pola yang lebih kompleks. Pada penelitian ini digunakan skema *many-to-one*,

dimana enam *timestep* data digunakan sebagai input *sequence* dan satu output digunakan sebagai target prediksi. *Hidden state* terakhir pada *layer 2* digunakan sebagai input pada output *layer* untuk menghasilkan prediksi jumlah kasus DBD.

Output Layer :

Perhitungan pada *sequence* pertama:

$$X_1 = \begin{bmatrix} 0.742 & 0.615 & 0.801 & 0.693 \\ 0.688 & 0.632 & 0.784 & 0.655 \\ 0.801 & 0.598 & 0.842 & 0.744 \\ 0.756 & 0.607 & 0.825 & 0.721 \\ 0.690 & 0.620 & 0.790 & 0.670 \\ 0.825 & 0.590 & 0.856 & 0.768 \end{bmatrix}$$

Karena *timestep* = 6:

$$Y_1 = 0.734$$

Dari hasil *forward propagation*:

$$h_6^{(2)} = \begin{bmatrix} 0.831 \\ 0.817 \\ 0.803 \\ 0.838 \end{bmatrix}$$

Persamaan output:

$$\hat{y} = W_o h_6^{(2)} + b_o$$

Diketahui:

$$W_o = [0.312 \quad 0.287 \quad 0.298 \quad 0.321]$$

$$b_o = 0.105$$

Langkah perhitungan:

$$\begin{aligned} \hat{y} &= (0.312)(0.831) + (0.287)(0.817) + (0.298)(0.803) + (0.321)(0.838) \\ &\quad + 0.105 \end{aligned}$$

$$\hat{y} = 0.259 + 0.234 + 0.239 + 0.269 + 0.105 = 1.106$$

Setelah nilai prediksi diperoleh, langkah selanjutnya adalah menghitung nilai *loss* menggunakan *Mean Squared Error* (MSE) pada persamaan (2.7), dengan nilai *loss function* dinotasikan sebagai L . Pada contoh perhitungan ini digunakan satu data pengamatan, sehingga jumlah data (n) bernilai 1. Karena nilai $n = 1$, faktor $\frac{1}{n}$ pada persamaan MSE menjadi 1, sehingga persamaan dapat disederhanakan menjadi kuadrat selisih antara nilai aktual (y) dan nilai prediksi ($\hat{y}$), yaitu:

$$L = (y - \hat{y})^2$$

Substitusi:

$$L = (0.734 - 1.106)^2$$

$$L = (-0.372)^2$$

$$L = 0.138$$

Perhitungan manual yang sudah dilakukan berlaku untuk *sequence* berikutnya yaitu X_2, X_3, X_4 , dan seterusnya. Kemudian setelah melakukan perhitungan pada GRU didapatkan gradien, sehingga bobot dan bias akan terus diperbarui menggunakan *optimizer* Adam.

Backpropagation Through Time (BPTT) :

Pada proses ini, dilakukan perhitungan gradien *error* terhadap parameter model berupa bobot dan bias. Berikut contoh perhitungan pada satu elemen bobot output w_{o11} dan b_o .

Gradien bobot (*backpropagation*).

Bobot yang dihitung:

$$w_{o11} = 0.312$$

Gradien diperoleh menggunakan *chain rule*:

$$\frac{\partial L}{\partial w_{o11}} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w_{o11}}$$

Turunan *loss* terhadap output:

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$$

$$\frac{\partial L}{\partial \hat{y}} = 2(1.106 - 0.734)$$

$$\frac{\partial L}{\partial \hat{y}} = 2(0.372) = 0.744$$

Turunan output terhadap bobot:

$$\hat{y} = w_{o11}h_{6,1}^{(2)} + b_o$$

$$\frac{\partial \hat{y}}{\partial w_{o11}} = h_{6,1}^{(2)}$$

$$\frac{\partial \hat{y}}{\partial w_{o11}} = 0.831$$

Sehingga gradien untuk w_{o11} :

$$\frac{\partial L}{\partial w_{o11}} = 0.744 \times 0.831$$

$$\frac{\partial L}{\partial w_{o11}} = 0.618$$

Parameter *optimizer* Adam yang digunakan adalah:

$$\text{Learning rate } (\eta) = 0.001$$

$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

$$\varepsilon = 1 \times 10^{-7}$$

1. Estimasi momen pertama pada persamaan (2.9)

Karena iterasi pertama:

$$m_0 = 0$$

maka estimasi momen pertama diperoleh dari gradien pada iterasi pertama.

$$m_1 = \beta_1 m_0 + (1 - \beta_1) g_1$$

$$m_1 = (0.9)(0) + (1 - 0.9)(0.618)$$

$$m_1 = 0 + 0.1(0.618)$$

$$m_1 = 0.0618$$

2. Estimasi momen kedua pada persamaan (2.10)

Karena iterasi pertama:

$$v_0 = 0$$

maka langkah perhitungan sebagai berikut.

$$v_1 = \beta_2 v_0 + (1 - \beta_2) g_1^2$$

$$v_1 = (0.999)(0) + (1 - 0.999)(0.618)^2$$

$$v_1 = 0.001(0.381924)$$

$$v_1 = 0.000382$$

3. *Bias correction* pada persamaan (2.11)

Momen pertama:

$$\hat{m}_1 = \frac{m_1}{1 - \beta_1^1}$$

$$\hat{m}_1 = \frac{0.0618}{1 - 0.9}$$

$$\hat{m}_1 = \frac{0.0618}{0.1} = 0.618$$

Momen kedua:

$$\hat{v}_1 = \frac{v_1}{1 - \beta_2^1}$$

$$\hat{v}_1 = \frac{0.000382}{1 - 0.999}$$

$$\hat{v}_1 = \frac{0.000382}{0.001} = 0.382$$

4. Memperbarui parameter bobot pada persamaan (2.12)

Pembaruan parameter dilakukan menggunakan:

$$w_{o11(new)} = w_{o11} - \eta \frac{\hat{m}_1}{\sqrt{\hat{v}_1} + \varepsilon}$$

$$w_{o11(new)} = 0.312 - 0.001 \left(\frac{0.618}{\sqrt{0.382} + 10^{-7}} \right)$$

$$w_{o11(new)} = 0.312 - 0.001 \left(\frac{0.618}{0.618} \right)$$

$$w_{o11(new)} = 0.312 - 0.001$$

$$w_{o11(new)} = 0.311$$

Gradien bias (*backpropagation*).

Bias yang dihitung:

$$b_o = 0.105$$

Gradien bias dihitung menggunakan *chain rule*:

$$\frac{\partial L}{\partial b_o} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial b_o}$$

Turunan *loss* terhadap output:

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y)$$

$$\frac{\partial L}{\partial \hat{y}} = 2(1.106 - 0.734)$$

$$\frac{\partial L}{\partial \hat{y}} = 2(0.372) = 0.744$$

Turunan output terhadap bias:

$$\hat{y} = w_{o11}h_{6,1}^{(2)} + b_o$$

$$\frac{\partial \hat{y}}{\partial b_o} = 1$$

Sehingga gradien untuk b_o :

$$\frac{\partial L}{\partial b_o} = 0.744 \times 1$$

$$\frac{\partial L}{\partial b_o} = 0.744$$

Parameter *optimizer* Adam yang digunakan sama dengan pembaruan bias.

1. Estimasi momen pertama pada persamaan (2.9)

Karena iterasi pertama:

$$m_0 = 0$$

maka estimasi momen pertama diperoleh dari gradien pada iterasi pertama.

$$m_1 = \beta_1 m_0 + (1 - \beta_1)g_1$$

$$m_1 = (0.9)(0) + (1 - 0.9)(0.744)$$

$$m_1 = 0.0744$$

2. Estimasi momen kedua pada persamaan (2.10)

Karena iterasi pertama:

$$v_0 = 0$$

maka langkah perhitungan sebagai berikut.

$$v_1 = \beta_2 v_0 + (1 - \beta_2)g_1^2$$

$$v_1 = (0.999)(0) + (1 - 0.999)(0.618)^2$$

$$v_1 = 0.000554$$

3. *Bias correction* pada persamaan (2.11)

Momen pertama:

$$\hat{m}_1 = \frac{m_1}{1 - \beta_1^1}$$

$$\hat{m}_1 = \frac{0.0744}{1 - 0.9} = 0.744$$

Momen kedua:

$$\hat{v}_1 = \frac{v_1}{1 - \beta_2}$$

$$\hat{v}_1 = \frac{0.000554}{1 - 0.999} = 0.554$$

4. Memperbarui parameter bobot pada persamaan (2.12)

Pembaruan parameter dilakukan menggunakan:

$$b_{o(new)} = b_o - \eta \frac{\hat{m}_1}{\sqrt{\hat{v}_1} + \varepsilon}$$

$$b_{o(new)} = 0.105 - 0.001 \left(\frac{0.744}{\sqrt{0.554} + 10^{-7}} \right)$$

$$b_{o(new)} = 0.105 - 0.001$$

$$b_{o(new)} = 0.104$$

Contoh perhitungan tersebut dilakukan pada satu parameter untuk mempermudah penjelasan. Pada implementasi model GRU sebenarnya, proses *backpropagation* dan pembaruan parameter menggunakan *optimizer* Adam dilakukan pada seluruh bobot dan bias pada setiap unit serta lapisan GRU secara berulang di setiap iterasi hingga proses pelatihan selesai.

Tabel 4.8 Parameter Arsitektur Model GRU (*Stacked* GRU)

No	Nama <i>Layer</i>	Output <i>Shape</i>	Parameter (Gate)	Total Parameter
1	<i>Layer 1</i>	(None, 64)	Reset Gate: 4.480 Update Gate: 4.480 Candidate: 4.480	13.440
2	<i>Dropout</i>	(None, 64)	Tidak ada parameter	0
3	<i>Layer 2</i>	(None, 64)	Reset Gate: 8.320 Update Gate: 8.320 Candidate: 8.320	24.960
4	<i>Dense</i>	(None, 1)	Tidak ada gerbang	65
			TOTAL PARAMETER	38.465

4.6 Hasil Prediksi Jumlah DBD

Hasil perbandingan antara nilai prediksi dan data aktual kasus Demam Berdarah Dengue (DBD) selanjutnya disajikan dalam Tabel 4.9 dan divisualisasikan dalam Gambar 4.6. Tabel tersebut menampilkan nilai hasil prediksi model serta nilai aktual pada periode pengamatan, sedangkan grafik pada Gambar 4.6 digunakan untuk memperlihatkan kesesuaian pola antara hasil prediksi dan data aktual secara visual. Melalui penyajian ini, kinerja model dalam memprediksi jumlah kasus DBD dapat diamati dengan lebih jelas, khususnya dalam menangkap tren dan pola perubahan data dari waktu ke waktu.

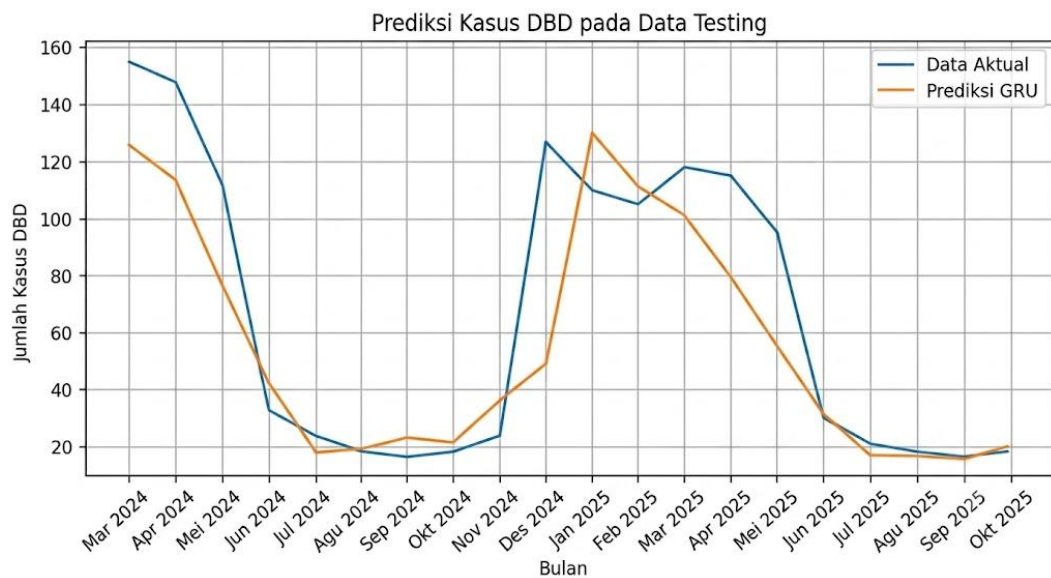
Tabel 4.9 Hasil Prediksi dan Data Aktual DBD

Bulan	Aktual DBD	Prediksi DBD
Mar 2024	155	126
Apr 2024	148	114
Mei 2024	112	75
Jun 2024	28	38
Jul 2024	18	12
Agu 2024	12	13
Sep 2024	10	18
Okt 2024	12	16
Nov 2024	18	32
Des 2024	127	46
Jan 2025	110	131
Feb 2025	105	112

Lanjutan Tabel 4.9 Hasil Prediksi dan Data Aktual DBD

Bulan	Aktual DBD	Prediksi DBD
Mar 2025	118	101
Apr 2025	115	78
Mei 2025	95	52
Jun 2025	25	27
Jul 2025	15	11
Agu 2025	12	11
Sep 2025	10	10
Okt 2025	12	14

Proses denormalisasi pada Tabel 4.9 dilakukan untuk mengembalikan nilai hasil prediksi ke skala data aslinya. Proses ini dilakukan dengan mengalikan hasil output model (data prediksi) serta data aktual menggunakan nilai maksimum yang diperoleh dari data pelatihan (*training set*). Penggunaan parameter data pelatihan secara konsisten pada kedua jenis data tersebut bertujuan untuk menjaga integritas skala data, sehingga perhitungan metrik evaluasi MAPE dapat memberikan gambaran tingkat akurasi yang valid.

**Gambar 4.6** Plot Hasil Prediksi dan Data Aktual

Berdasarkan Gambar 4.6, model GRU terbukti mampu mengikuti pola fluktuasi data aktual kasus DBD secara signifikan. Model berhasil menangkap momentum perubahan tren, terutama pada lonjakan tajam di observasi ke-4 hingga ke-8 (dari kisaran 15 hingga puncak 155 kasus). Begitu pula pada penurunan drastis dari observasi ke-9 menuju ke-11 (dari 148 ke 28 kasus), kurva prediksi menunjukkan pergerakan yang searah dan identik dengan data riil. Meskipun terdapat selisih nilai (*error*) pada beberapa titik ekstrem, kemampuan model dalam mempelajari pola kenaikan dan penurunan ini menunjukkan bahwa GRU memiliki presisi visual yang memadai dalam memprediksi tren kasus DBD di Kota Malang.

4.7 Evaluasi Model

Evaluasi model dilakukan menggunakan *Mean Absolute Percentage Error* (MAPE) pada data testing untuk mengukur tingkat akurasi prediksi masing-masing metode yang digunakan dalam penelitian. Semakin kecil nilai MAPE yang dihasilkan, maka semakin baik kemampuan model dalam memprediksi jumlah kasus Demam Berdarah Dengue (DBD). Tabel 4.10 menunjukkan perbandingan nilai MAPE dari model SARIMAX dan *Gated Recurrent Unit* (GRU).

Tabel 4.10 Perbandingan Nilai MAPE Model Prediksi DBD

Model	MAPE (%)
SARIMAX(1,1,1)(1,1,0) ₁₂	21.58
<i>Gated Recurrent Unit</i> (GRU)	29.29

Berdasarkan Tabel 4.10, model SARIMAX(1,1,1)(1,1,0)₁₂ menghasilkan nilai MAPE sebesar 21.58%, sedangkan model *Gated Recurrent Unit* (GRU) menghasilkan nilai MAPE sebesar 29.29%. Karena nilai MAPE yang lebih kecil menunjukkan tingkat kesalahan prediksi yang lebih rendah, maka model

SARIMAX memiliki performa prediksi yang lebih baik dibandingkan model GRU dalam memprediksi jumlah kasus DBD di Kota Malang.

Hasil tersebut menunjukkan bahwa pola jumlah kasus DBD pada data penelitian memiliki karakteristik musiman yang kuat sehingga dapat ditangkap dengan baik oleh model SARIMAX melalui komponen *autoregressive*, *moving average*, dan musiman. Sebaliknya, meskipun GRU mampu mempelajari hubungan nonlinier antara jumlah kasus DBD dan variabel iklim, performanya masih berada di bawah SARIMAX. Hal ini dapat disebabkan oleh jumlah data yang terbatas sehingga proses pembelajaran GRU belum optimal. Selain itu, kompleksitas model GRU yang lebih tinggi memerlukan jumlah data yang lebih besar untuk memperoleh pola stabil, sedangkan SARIMAX lebih efektif memodelkan data deret waktu dengan pola musiman yang jelas pada ukuran data yang relatif kecil.

4.8 Integrasi Hasil Penelitian dengan Kajian Agama

Landasan filosofis dari penelitian ini berakar pada perintah Allah SWT dalam QS. Yunus ayat 101 yang menginstruksikan manusia untuk memperhatikan (*unzhuru*) segala fenomena yang terjadi di langit dan di bumi. Perintah ini merupakan seruan bagi hamba-Nya untuk melakukan observasi mendalam terhadap keteraturan alam semesta sebagai tanda-tanda kekuasaan-Nya. Manifestasi dari observasi tersebut dijabarkan lebih rinci dalam QS. Al-Baqarah ayat 164, yang menegaskan bahwa fenomena alam seperti pergantian malam dan siang, serta turunnya hujan yang menghidupkan bumi yang mati, bukanlah kejadian acak melainkan sebuah sistem yang terukur (*sunnatullah*). Dalam konteks epidemiologi, siklus air dan dinamika iklim yang disebutkan dalam ayat tersebut merupakan

variabel deterministik yang memengaruhi ekosistem perkembangbiakan vektor nyamuk *Aedes aegypti*. Oleh karena itu, fluktuasi jumlah kasus Demam Berdarah Dengue (DBD) yang menjadi objek penelitian ini pada hakikatnya adalah data empiris yang mencerminkan interaksi kompleks antar berbagai tanda kekuasaan Allah di alam semesta.

Upaya manusia untuk menyingkap pola di balik fenomena tersebut dilakukan melalui penerapan teknologi kecerdasan buatan, khususnya algoritma *Gated Recurrent Unit* (GRU). Mekanisme *update gate* dan *reset gate* pada GRU berfungsi sebagai jaringan saraf buatan untuk menyaring informasi penting dari masa lalu, yang secara metodologis merupakan bentuk “perhatian” manusia terhadap data historis yang telah disediakan oleh alam. Berdasarkan hasil pengujian yang dilakukan dalam penelitian ini, model GRU menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) sebesar 29.29%. Angka ini menunjukkan bahwa model memiliki kemampuan prediksi yang berada pada kategori cukup. Meskipun terdapat margin *error* sebesar 29.29%, hasil ini membuktikan bahwa akal manusia mampu memetakan tren besar penyebaran penyakit yang dipengaruhi oleh dinamika lingkungan. Adanya selisih antara nilai prediksi dan aktual tersebut sekaligus menjadi pengingat akan keterbatasan manusia dalam menangkap seluruh variabel ilahiah yang maha luas. Namun demikian, nilai akurasi yang dicapai tetap memberikan kontribusi praktis yang signifikan sebagai sistem peringatan dini bagi masyarakat, selaras dengan tujuan syariat untuk menjaga keselamatan jiwa (*hifdzun nafs*) melalui ikhtiar saintifik yang terukur.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan sebagai berikut:

1. Berdasarkan hasil analisis dan pengujian pada Tabel 4.9, dapat disimpulkan bahwa penerapan model *Gated Recurrent Unit* (GRU) yang mengintegrasikan variabel *Lag* Kasus DBD serta unsur iklim (curah hujan, suhu, dan kelembapan) terbukti efektif dalam memprediksi jumlah kasus DBD di Kota Malang. Secara sistematis, model mampu menangkap pola hubungan antara data masa lalu dengan data aktual. Efektivitas ini ditunjukkan oleh kemampuan model dalam mengikuti fluktuasi tren kasus yang cukup dinamis, seperti pada periode Juni 2024 hingga Oktober 2024, model berhasil mendeteksi penurunan kasus secara drastis hingga ke angka belasan. Begitu pula pada periode Juni 2025 hingga Oktober 2025, hasil prediksi menunjukkan akurasi yang sangat tinggi dengan selisih yang mendekati nilai data aktual.
2. Model terbaik dalam penelitian ini adalah SARIMAX(1,1,1)(1,1,0)₁₂ dengan nilai MAPE sebesar 21.58%, sehingga lebih baik dibandingkan model GRU yang menghasilkan nilai MAPE sebesar 29.29%. Hal ini menunjukkan bahwa pola musiman pada data kasus DBD lebih dominan dan lebih efektif dimodelkan menggunakan pendekatan SARIMAX dibandingkan pendekatan *deep learning* GRU pada data yang digunakan dalam penelitian ini.

5.2 Saran

Beberapa saran yang diberikan untuk penelitian selanjutnya adalah memperluas rentang waktu dataset guna mendalami pola musiman dan meningkatkan akurasi prediksi. Selain faktor iklim, eksplorasi variabel tambahan seperti kepadatan penduduk dan faktor lingkungan fisik lainnya perlu dilakukan karena pengaruhnya terhadap penyebaran nyamuk *Aedes aegypti*. Saran terakhir yaitu optimasi *hyperparameter* pada jumlah lapisan, *learning rate*, dan *batch size* diperlukan untuk meminimalkan nilai *error* dan meningkatkan presisi hasil.

DAFTAR PUSTAKA

- Agustin, I., Tarwotjo, U., & Rahadian, R. (2017). Perilaku Bertelur dan Siklus Hidup *Aedes Aegypti* pada Berbagai Media Air. *Jurnal Biologi*, 6(4), 71-81.
- Alkahfi, C., Kurnia, A., & Saefuddin, A. (2024). Performance Comparison of RNN-Based Models in Forecasting Indonesian Economic and Financial Data. *MALCOM: Indonesian Journal of machine Learning and Computer Science*, 4(4), 1235-1243.
- Aksan, F., Suresh, V., Janik, P., & Sikorski, T. (2023). Load Forecasting for the Laser Metal Processing Industry Using VMD and Hybrid Deep Learning Models. *Energies*, 16, 5381. <https://doi.org/10.3390/en16145381>.
- Arunkumar, K, E., Kalaga, D, V., Kumar, C, M., Kawaji, M., Brenza, M. (2022). Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends. *Alexandria Engineering Journal*, 61, 7585-7603. <https://doi.org/10.1016/j.aej.2022.01.011>.
- Bartz, E., Bartz-Beielstein, T., Zaefferer, M., & Mersmann, O. (Eds.). (2023). Hyperparameter tuning for machine and deep learning with R: A practical guide. Springer.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.
- Dermawan, H., Yuliana, M., & Hadi, M, Z. (2023). GRU and XGBoost Performance with Hyperparameter Tuning Using GridSearchCV and Bayesian Optimization on an IoT-Based Weather Prediction System. *International Journal on Advanced Science, Engineering and Information Technology*, 13(3), 851-860.
- Dinkes Kota Malang. (2025). *Demam Berdarah Dengue*. <https://dinkes.malangkota.go.id/tag/dbd/>
- El-Meehy, A, O., El-Kharbotly, A, K., & El-Beheiry, M, M. (2025). Systematic hyperparameter analysis of GRU and LSTM across demand pattern types: a demandcharacteristic-driven meta-learning framework for rapid optimization. *Scientific Reports*, 15:44657. <https://doi.org/10.1038/s41598-025-31508-x>.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

- Gema, B., & Suhartono. (2020). *Perbandingan Kinerja LSTM dan GRU dalam Peramalan Deret Waktu Medis*. Jurnal Ilmu Komputer dan Informatika, 8(2), 123–132.
- Kahar, F. P., Abidin, A., & Ilham, R. (2024). Analisis Tingkat Intensitas Curah Hujan, Tekanan Udara serta Suhu Udara di Wilayah Paotere Makassar Selama Periode Tahun 2022. *Jurnal Sains Fisika*, 4(1), 27-36.
- Kemenag. (2024). *Qur'an Kemenag*.
<https://quran.kemenag.go.id/quran/perayat/surah/2?from=195&to=286>.
- Kemkes RI. (2024). *Laporan Kasus DBD*. Diakses pada 9 Juni 2025 dari
<https://ayosehat.kemkes.go.id/topik/demam-berdarah-dengue>.
- Khairunisa, N. K., & Hendikawati, P. (2024). Long Short-Term Memory and Gated Recurrent Unit Modeling For Stock Price Forecasting. *JMSK*, 21(1), 321–333. <https://10.20956/j.v21i1.35930>.
- Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*. arXiv:1412.6980.
- Lahdji, A., & Putra, B. B. (2017). Hubungan Curah Hujan, Suhu, Kelembaban dengan Kasus Demam Berdarah Dengue di Kota Semarang. *Syifa Medika: Jurnal Kedokteran dan Kesehatan*, 8(1), 46-53.
- Lestari, P. A., Fajar, N. A., Windusari, Y., Novrikasari, & Sunarsih, E. (2023). Faktor Pengaruh Kesehatan Lingkungan terhadap Kejadian Demam BerdarahDangue(DBD) di Wilayah Endemis: SystematicLiterature Review. *Health Information : Jurnal Penelitian*, 15(3), 1-10.
- Lin, H., Gharehbaghi, A., Zhang, Q., Band, S. S., Pai, H. T., Chau, K., & Mosavi, A. (2022). Time series-based groundwater level forecasting using gated recurrent unit deep neural networks. *Engineering Applications Of Computational Fluid Mechanics*, 16(1), 1655–1672. <https://doi.org/10.1080/19942060.2022.2104928>.
- Makruf, S. M., Ramadhani, L. R., Sandha, F., Rini, P. A., Najwa, S. A., Amrullah, A. H., Anatasa, S., & Prastia, A. S. (2025). Analisis Kelembaban Udara terhadap Tingginya Suhu di Sekaran Semarang. *Jurnal Analisis*, 4(1), 93-100.
- Marvianto, D., Ratih, O. D., & Wijaya, K. F. N. (2023). Infeksi Dengue Sekunder: Patofisiologi, Diagnosis, dan Implikasi Klinis. *CDK-313*, 50(2), 70-74.
- Melina, M., Sukono, Napitupulu, H., Mohamed, N., Chrisnanto, Y. H., Hadiana, A. I., & Kusumaningtyas, V. A. (2025). Cryptocurrency time series forecasting model using GRU algorithm based on machine learning. *Barekeng: Jurnal Ilmu Matematika dan Terapan*, 19(2), 1317–1328.

- Meriani, A. P., & Rahmatulloh, A. (2024). Perbandingan Gated Recurrent Unit (GRU) dan Algoritma Long Short Term Memory (LSTM) Linear Refression Dalam Prediksi Harga Emas Menggunakan Model Time Series. *Jurnal Informasi dan Teknik Elektro Terapan*, 12(1), 468-474.
- Nasir, M., Aldillah, S., Tenrisau, D., Haris, I, M., Rahastri, A., Sa'adatar, R. N., Surya, A., Thohir, B., Aryani, D., Firdaus, K, M. (2024). Machine Learning Approach to Predict the Dengue Cases Based on Climate Factors. *Window Of Health: Jurnal Kesehatan*, 7(2), 203-14.
- Noh, S. H. (2021). Analysis of Gradient Vanishing of RNNs and Performance Comparison. *Information*, 12, 442. <https://doi.org/10.3390/info12110442>.
- Nugraheni, E., Rizqoh, D., & Sundari, M. (2023). Manifestasi Klinis Demam Berdarah Dengue (DBD). *Jurnal Kedokteran dan Kesehatan: Publikasi Ilmiah Fakultas Kedokteran Universitas Sriwijaya*, 10(3), 267-274.
- Othman, M., Kusumo, R., & Abdullah, S. (2022). Model forecasting for dengue fever incidence using time-series analysis. *Processes*, 10(11), 2454.
- Rakhmatsani, L., & Susanna, D. (2024). Studi Ekologi Hubungan Iklim Terhadap Kejadian Demam Berdarah Dengue (DBD) di Kabupaten Bogor Tahun 2013-2022. *Jurnal Kesehatan Lingkungan Indonesia*, 23(2), 207-214. <https://doi.org/10.14710/jkli.23.2.207-214>.
- Riswanto, R. L., Sutikno, & Indriyati. (2014). Aplikasi Prediksi Jumlah Penderita Penyakit Demam Berdarah Dengue di Kota Semarang Menggunakan Jaringan Syaraf Tiruan Backpropagation. *Jurnal Masyarakat Informatika*, 5(10), 19-27.
- Rochman, E. M., Miswanto, Suprajitno, H., Rachmad, A., Nindyasari, R., Rachman, F. H. (2022). Perbandingan LSTM dan GRU dalam Memprediksi Jumlah Penderita Diabetes. *2022 IEEE 8th Information Technology International Seminar (ITIS)*, 145-149.
- Rodrigues, F., & Machado, M. (2025). High-Frequency Cryptocurrency Price Forecasting Using Machine Learning Models: A Comparative Study. *Information*, 16, 300. <https://doi.org/10.3390/info16040300>.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/323533a0>.
- Schaefer, T. J, Panda, P. K., & Wolford, R. W. (2021). *Dengue Fever*. Treasure Island (FL): StatPearls Publishing; 2021 Jan.

- Simmons, C. P., Ph. D., Jeremy J. Farrar, M.D., Ph.D., Nguyen van Vinh Chau, M.D., Ph.D., & Bridget Wills, M.D., D.M. Dengue. (2012). *The New England Journal of Medicine*, 366(15).
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). *Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research*, 15(1), 1929–1958.
- Tu, B., Bai, K., Zhan, C., & Zhang, W. (2024). Real-time prediction of ROP based onGRU-Informer. *Scientific Resports*, 34:2133.
<https://doi.org/10.1038/s41598-024-52261-7>.
- Umar, M. S., Tadeus, D. Y., Mangkusasmito, F., & Putranto, A. B. (2022). Rancang Bangun Sistem Pengendalian Suhu dan Kelembaban Pada Box Penyimpanan Produk Berbahan Kulit Berbasis Mikrokontroller Arduino Uno. *Berkala Fisika*, 25(1), 27-35.
- Yang, Y., Su, Z., Zhang, Y., Goh, C. C., Lin, Y., Bellotti, A. G., & Lee, B. G. (2025). Kolmogorov–Arnold Networks-based GRU and LSTM for Loan Default Early Prediction. arXiv preprint arXiv:2507.13685.
<https://arxiv.org/pdf/2507.13685.pdf>.
- Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics Conference Series*, 1168(2), 1-6. 10.1088/1742-6596/1168/2/022022.
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). Dive into deep learning. Cambridge University Press.
- Zhou, S., Yang, Z., Yu, Q., & Wang, J. (2025). Grid-Search-Optimized, Gated Recurrent Unit-Based Prediction Model for Ionospheric Total Electron Content. *Technologies* 2025, 13(8), 347.
<https://doi.org/10.3390/technologies13080347>.

LAMPIRAN

Lampiran 1. Data Jumlah Kasus DBD dan Faktor Iklim di Kota Malang

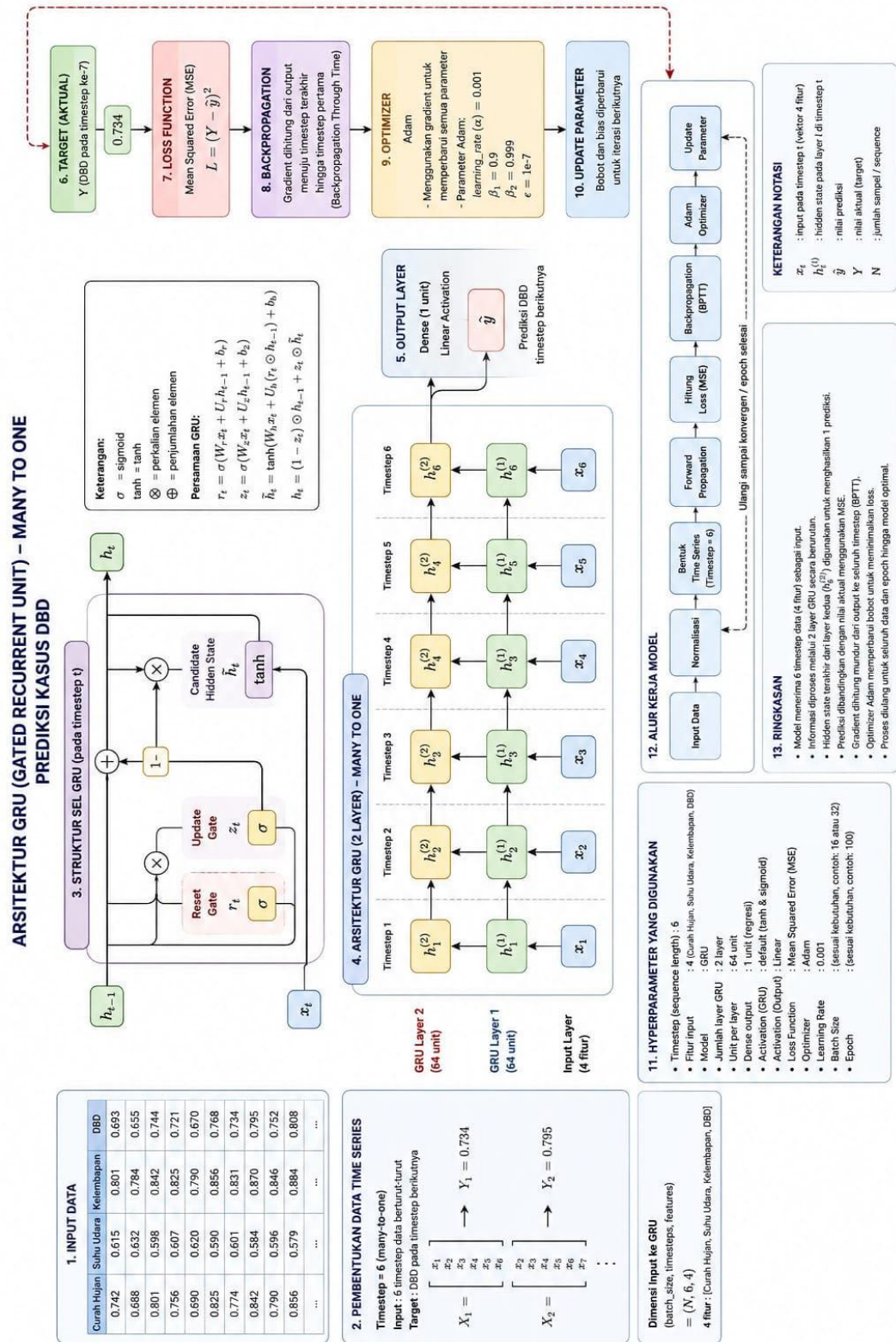
Waktu	DBD	Curah Hujan	Suhu Udara	Kelembapan Udara
Jan-15	52	750,14	23,66	85,13
Feb-15	58	672,86	23,8	83,67
Mar-15	62	581,43	24,18	83,03
Apr-15	55	10,24	24,01	83,77
May-15	38	1712,34	23,59	76,68
Jun-15	12	636,23	22,74	75,63
Jul-15	8	683,69	22,25	73,62
Aug-15	5	306,48	22,48	72,39
Sep-15	4	0,34	23,54	67,97
Oct-15	6	0	25,35	62,03
Nov-15	10	598,05	25,57	73,72
Dec-15	48	293,51	24,78	83
Jan-16	78	599,46	25,08	80,19
Feb-16	85	636,19	24,02	87
Mar-16	92	1232,41	24,97	83,6
Apr-16	88	955,16	25,12	80,62
May-16	62	601,2	25,06	80,35
Jun-16	18	311,92	24,22	80,37
Jul-16	12	577,04	23,72	79,32
Aug-16	8	576,33	23,57	76,65
Sep-16	7	319,02	24,56	76,2
Oct-16	9	643,7	24,81	77,9
Nov-16	15	14,46	24,66	83,33
Dec-16	70	1442,2	24,23	84,23
Jan-17	18	585,49	23,86	85,54
Feb-17	20	325,93	24,16	82,43
Mar-17	22	299,82	24,05	82,47
Apr-17	19	601,01	24,23	82,67
May-17	12	297,67	24,21	76,45
Jun-17	4	1,17	23,6	78,27
Jul-17	2	446,37	22,57	79,16
Aug-17	3	0,1	22,63	76,26
Sep-17	3	7,65	23,63	73,3
Oct-17	1	7,45	23,82	77,03
Nov-17	2	14,66	24,29	83,23
Dec-17	9	327,3	24,23	83,16
Jan-18	14	310,91	23,79	85,29
Feb-18	16	642,93	23,69	84,32

Waktu	DBD	Curah Hujan	Suhu Udara	Kelembapan Udara
Mar-18	17	619,83	24,04	82,03
Apr-18	15	595,2	24,79	77
May-18	10	0,6	24,14	74,97
Jun-18	3	1,87	23,16	78,07
Jul-18	2	0,32	21,94	75
Aug-18	1	0	22,16	73,26
Sep-18	1	0,11	23,78	70,67
Oct-18	1	0,21	25,15	69,35
Nov-18	2	1196,1	25,02	79,1
Dec-18	10	309,81	24,13	85,65
Jan-19	85	898,72	24,44	82,35
Feb-19	92	329,45	24,13	83,96
Mar-19	98	11,94	23,85	83,94
Apr-19	95	306,86	24,42	82,8
May-19	78	2,42	24,11	77,03
Jun-19	22	0	22,61	74,07
Jul-19	15	298,21	21,96	75,39
Aug-19	10	0	22,34	71,71
Sep-19	8	0	23,49	68,3
Oct-19	10	0,04	25,45	65,48
Nov-19	14	299,98	25,78	70,97
Dec-19	80	10,28	25,32	79,23
Jan-20	50	297,27	24,66	82,16
Feb-20	55	17,32	24,22	84,72
Mar-20	60	7,7	24,54	82,84
Apr-20	58	301,41	24,95	79,5
May-20	42	577,8	24,65	80,23
Jun-20	12	0,11	23,72	75,3
Jul-20	8	287,27	23,02	75,94
Aug-20	5	388,84	23,42	73,97
Sep-20	4	0,53	25,56	69,43
Oct-20	5	290,93	24,81	75,58
Nov-20	8	7,77	25,21	78,2
Dec-20	47	1443,85	23,85	84,03
Jan-21	32	12,05	23,72	85,1
Feb-21	35	334,8	24,26	81,71
Mar-21	38	296,38	23,89	83,55
Apr-21	36	300,39	24,46	76,7
May-21	25	891,56	24,58	76,57
Jun-21	8	599,41	24,15	78,5
Jul-21	5	0,78	23,08	72,88
Aug-21	4	1,15	23,62	76,31

Waktu	DBD	Curah Hujan	Suhu Udara	Kelembapan Udara
Sep-21	3	3,45	24,08	77,52
Oct-21	4	290,99	24,58	78,52
Nov-21	6	12,24	24,24	83,9
Dec-21	32	1153,76	25,33	80,97
Jan-22	95	580,39	24,2	81,81
Feb-22	105	342,23	23,96	83,25
Mar-22	112	584,67	24,29	83,06
Apr-22	108	12,23	24,38	78,87
May-22	85	862,54	24,88	77,9
Jun-22	20	899,07	23,6	80,47
Jul-22	12	0,7	23,33	75,61
Aug-22	8	614,47	23,5	76,7
Sep-22	6	4,37	24,06	78,73
Oct-22	7	302,71	23,75	84,77
Nov-22	10	15,79	23,98	85,07
Dec-22	92	7,89	24,49	79,32
Jan-23	72	8,98	24,32	80,29
Feb-23	68	7,5	23,84	83,96
Mar-23	75	8,93	24,11	80,16
Apr-23	70	3,84	24,08	82,77
May-23	58	3,42	24,44	74,06
Jun-23	18	332,27	23,88	78,37
Jul-23	12	987,66	22,68	77,87
Aug-23	10	329,2	22,78	73,03
Sep-23	8	370,33	23,73	68
Oct-23	10	444,56	26,08	65,74
Nov-23	15	722,91	26,49	73,1
Dec-23	66	12,1	25,98	75,87
Jan-24	135	619,58	25,68	80,19
Feb-24	142	355,88	25,63	81,66
Mar-24	155	305,46	25,95	79,39
Apr-24	148	353,39	26,39	79,03
May-24	112	0,09	24,44	72,13
Jun-24	28	0,98	25,56	73,77
Jul-24	18	306,61	23,44	72,93
Aug-24	12	2,37	25,31	73,06
Sep-24	10	890,28	24,62	72,23
Oct-24	12	865,85	25,58	72,03
Nov-24	18	893,93	25,76	77,87
Dec-24	127	298,25	24,45	87,03
Jan-25	110	298,84	24,23	85,29
Feb-25	105	327,76	24,7	82,14

Waktu	DBD	Curah Hujan	Suhu Udara	Kelembapan Udara
Mar-25	118	580,77	24,69	82,55
Apr-25	115	298,82	24,69	79,77
May-25	95	1153,35	24,65	82,16
Jun-25	25	594,22	24,06	79,33
Jul-25	15	286,83	23,08	78,45
Aug-25	12	577,05	23,45	77,84
Sep-25	10	2,59	24,38	76,53
Oct-25	12	8,18	25,05	77,55

Lampiran 2. Arsitektur GRU untuk Prediksi Kasus DBD



Lampiran 3. Syntax Prediksi Kasus DBD Menggunakan Metode GRU

```

from google.colab import files
uploaded = files.upload()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_percentage_error, mean_absolute_error,
mean_squared_error

data = pd.read_excel("DATA SKRIPSI.xlsx")
data.head()

if 'Waktu' in data.columns:
    data['Waktu'] = pd.to_datetime(data['Waktu'])
    data = data.set_index('Waktu')

stades_asli = data.describe()
print(stades_asli)

plt.figure(figsize=(12,6))

for col in data.columns:
    plt.plot(data[col], label=col)

plt.title("Plot Data")
plt.legend()
plt.show()

train_size = int(len(data)*0.8)

train = data.iloc[:train_size]
test = data.iloc[train_size:]

print("Jumlah data training:",len(train))
print("Jumlah data testing:",len(test))

scaler = MinMaxScaler()

train_scaled = scaler.fit_transform(train)
test_scaled = scaler.transform(test)

train_scaled = pd.DataFrame(train_scaled, columns=data.columns)
test_scaled = pd.DataFrame(test_scaled, columns=data.columns)

print("Statistika Deskriptif Data Training Normalisasi")
print(train_scaled.describe())

print("Statistika Deskriptif Data Testing Normalisasi")

```

```

print(test_scaled.describe())

def create_dataset(dataset, timestep):

    X=[]
    y=[]

    for i in range(timestep,len(dataset)):
        X.append(dataset[i-timestep:i])
        y.append(dataset[i,0])

    return np.array(X), np.array(y)

timestep = 6

X_train,y_train = create_dataset(train_scaled.values,timestep)
X_test,y_test = create_dataset(test_scaled.values,timestep)

units_list = [32,64]
layer_list = [1,2]
lr_list = [0.001]
dropout_list = [0.1,0.2]
epoch_list = [100]
batch_list = [8,16]

best_mape = 999
best_model = None
best_param = None

for units in units_list:

    for layers in layer_list:

        for lr in lr_list:

            for drop in dropout_list:

                for batch in batch_list:

                    model = Sequential()

                    if layers==1:

                        model.add(GRU(units,
                                      input_shape=(X_train.shape[1],X_train.shape[2])))

                        model.add(Dropout(drop))

                    if layers==2:

                        model.add(GRU(units,
                                      return_sequences=True,
                                      input_shape=(X_train.shape[1],X_train.shape[2])))

                        model.add(Dropout(drop))

                        model.add(GRU(units))

```

```

model.add(Dense(1))

optimizer = Adam(learning_rate=lr)

model.compile(
    optimizer=optimizer,
    loss='mse')

early_stop = EarlyStopping(
    monitor='loss',
    patience=20,
    restore_best_weights=True)

model.fit(
    X_train,
    y_train,
    epochs=100,
    batch_size=batch,
    verbose=0,
    callbacks=[early_stop])

pred = model.predict(X_test)

temp = np.zeros((len(pred),data.shape[1]))
temp[:,0] = pred[:,0]

pred_denorm = scaler.inverse_transform(temp)[:,0]

temp2 = np.zeros((len(y_test),data.shape[1]))
temp2[:,0] = y_test

y_denorm = scaler.inverse_transform(temp2)[:,0]

mape = mean_absolute_percentage_error(y_denorm,pred_denorm)*100

print("Units:",units,
      "Layers:",layers,
      "LR:",lr,
      "Dropout:",drop,
      "Batch:",batch,
      "MAPE:",mape)

if mape < best_mape:

    best_mape = mape
    best_model = model
    best_param = (units,layers,lr,drop,batch)

history = model.fit(
    X_train,
    y_train,
    epochs=100,
    batch_size=batch,
    verbose=1)

best_loss = min(history.history['loss'])

print("Loss terbaik:", best_loss)

```

```

best_epoch = np.argmin(history.history['loss']) + 1

print("Epoch terbaik:", best_epoch)
print("Loss terbaik:", best_loss)

print("Hyperparameter terbaik")

print("Units:",best_param[0])
print("Layer:",best_param[1])
print("Learning rate:",best_param[2])
print("Dropout:",best_param[3])
print("Batch size:",best_param[4])
print("MAPE terbaik:",best_mape)

pred = best_model.predict(X_test)

temp = np.zeros((len(pred),data.shape[1]))
temp[:,0] = pred[:,0]

pred_denorm = scaler.inverse_transform(temp)[:,0]

temp2 = np.zeros((len(y_test),data.shape[1]))
temp2[:,0] = y_test

y_denorm = scaler.inverse_transform(temp2)[:,0]

hasil = pd.DataFrame({
"Aktual":y_denorm,
"Prediksi":pred_denorm})

print(hasil)

mape = mean_absolute_percentage_error(y_denorm,pred_denorm)*100
mae = mean_absolute_error(y_denorm,pred_denorm)
rmse = np.sqrt(mean_squared_error(y_denorm,pred_denorm))

print("MAPE:",mape)
print("MAE:",mae)
print("RMSE:",rmse)

plt.figure(figsize=(10,5))

plt.plot(y_denorm,label="Data Aktual")
plt.plot(pred_denorm,label="Prediksi GRU")

plt.title("Prediksi Kasus DBD")
plt.legend()

plt.show()

```


DAFTAR RIWAYAT HIDUP



Silvi Nur Afifa Darmayanti lahir di Malang pada tanggal 03 Maret 2004. Penulis merupakan anak bungsu dari dua bersaudara, putri dari pasangan Ayahanda Darmaji dan Ibu Dami Ariyanti. Pendidikan formal dimulai di SDN Bedali 01 dan lulus pada tahun 2016, kemudian melanjutkan jenjang pendidikan menengah pertama di MTsN 3 Malang hingga lulus pada tahun 2019. Selanjutnya, penulis menempuh pendidikan menengah atas di SMAN 1 Singosari dan lulus pada tahun 2022. Setelah itu, penulis melanjutkan studi di Universitas Islam Negeri Maulana Malik Ibrahim Malang pada Program Studi Matematika, Fakultas Sains dan Teknologi. Selama perkuliahan, penulis mengikuti sejumlah organisasi lain, baik intra maupun ekstra kampus yang dapat mendukung pengembangan diri penulis. Penulis juga pernah melaksanakan kegiatan magang di Badan Pusat Statistik (BPS) Kota Probolinggo sebagai bagian dari pemenuhan program Praktik Kerja Lapangan (PKL) universitas. Pengalaman tersebut menjadi bekal berharga dalam membentuk karakter, memperluas wawasan, serta mempersiapkan diri untuk berkontribusi secara profesional maupun sosial.



KEMENTERIAN AGAMA RI

UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM

MALANG FAKULTAS SAINS DAN TEKNOLOGI

Jl. Gajayana No. 50 Malang 65144 Telp. / Fax. (0341) 558933

BUKTI KONSULTASI SKRIPSI

Nama : Silvi Nur Afifa Darmayanti
NIM : 220601110099
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Penerapan *Gated Recurrent Unit* (GRU) Untuk Prediksi
Jumlah Kasus Demam Berdarah Dengue (DBD) (Studi Kasus: Kota Malang)
Pembimbing I : Ria Dhea Layla Nur Karisma, M.Si
Pembimbing II : Mohammad Nafie Jauhari, M.Si

No	Tanggal	Hal	Tanda Tangan
1.	13 Juni 2025	Konsultasi Topik	1.
2.	10 September 2025	Konsultasi Bab I	2.
3.	18 September 2025	Konsultasi Bab I, II, III	3.
4.	25 September 2025	Konsultasi Bab I, II, III	4.
5.	1 Oktober 2025	Konsultasi Bab I, II, III	5.
6.	8 Oktober 2025	Konsultasi Bab I, II, III	6.
7.	15 Oktober 2025	Konsultasi Bab I, II, III	7.
8.	22 Oktober 2025	Konsultasi Bab I, II, III	8.
9.	29 Oktober 2025	Konsultasi Bab I, II, III	9.
10.	30 Oktober 2025	Konsultasi Kajian Agama Bab I dan II	10.
11.	31 Oktober 2025	ACC Kajian Agama Bab I dan II	11.
12.	5 November 2025	ACC Bab I, II, III	12.
13.	11 November 2025	ACC Seminar Proposal	13.



KEMENTERIAN AGAMA RI

UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM

MALANG FAKULTAS SAINS DAN TEKNOLOGI

Jl. Gajayana No. 50 Malang 65144 Telp. / Fax. (0341) 558933

14.	9 Desember 2025	Konsultasi Revisi Seminar Proposal	14.
15.	3 Februari 2026	Konsultasi Bab IV dan V	15.
16.	12 Februari 2026	Konsultasi Bab IV dan V	16.
17.	24 Februari 2026	Konsultasi Bab IV dan V	17.
18.	3 Maret 2026	Konsultasi Bab IV dan V	18.
19.	10 Maret 2026	Konsultasi Bab IV dan V	19.
20.	2 April 2026	ACC Kajian Agama Bab IV	20.
21.	7 April 2026	ACC Bab IV dan V	21.
22.	9 April 2026	ACC Seminar Hasil	22.
23.	5 Mei 2026	Konsultasi Revisi Seminar Hasil	23.
24.	19 Mei 2026	ACC Sidang Skripsi	24.
25.	26 Mei 2026	ACC Keseluruhan	25.

Malang, 26 Mei 2026

Mengetahui,

Kepa Program Studi Matematika



Dr. Fachrur Rozi, M.Si

NIP. 19800527 200801 1 012