

BAB VI

PENUTUP

6.1. Simpulan

Berdasarkan hasil penelitian, implementasi sistem, serta seluruh rangkaian pengujian fungsional yang dikelola secara iteratif menggunakan metodologi *Scrum*, diperoleh kesimpulan yang menjawab seluruh rumusan masalah penelitian sebagai berikut:

1. Integrasi API dan Otomatisasi Data: Penelitian ini telah berhasil merancang dan membangun arsitektur Sistem Informasi *Monitoring Near-Earth Objects* (SINEO) menggunakan *framework* Laravel 12 yang terintegrasi secara otomatis dengan REST API NASA CNEOS melalui otentikasi kunci API resmi. Sistem hibrida ini mampu mengotomatisasi penarikan riwayat elemen orbital asteroid secara *real-time* ke pangkalan data relasional MySQL, sehingga berhasil meningkatkan efisiensi tata kelola data astronomi serta memangkas latensi pemuatan antarmuka melalui penerapan strategi penyimpanan memori jangka pendek (*caching*) lokal pada server.
2. Kinerja Prediksi *Recurrent Neural Network* (RNN): Penerapan algoritma *Deep Learning* dengan model sekuensial *Dual-Stacked Recurrent Neural Network* (RNN) berbasis bahasa pemrograman *Python* terbukti andal dalam memproses data historis elemen orbital asteroid. Pengujian pengolahan data runtun waktu (*time-series*) sepanjang 100 *epoch* penuh berhasil menekan nilai fungsi kerugian kuadrat rata-rata hingga menyentuh titik konvergensi *Mean Squared Error* (MSE) murni sebesar 0.000537. Menggunakan pendekatan evaluasi *Root Mean Squared Error* (RMSE) yang dikalibrasi dengan faktor skala varians data sebesar 3.5, model menghasilkan tingkat akurasi konvergensi bobot yang objektif pada *normalized space* sebesar 91.9%. Model ini teruji valid untuk memproyeksikan estimasi fluktuasi nilai jarak meleset (*miss_distance_km*) secara rekursif (*Recursive Multi-step Forecasting*) untuk sepekan (7 hari) ke depan secara presisi dan terbebas dari bias orientasi (*overfitting*).

3. Penyederhanaan Visualisasi Antarmuka Dashboard: Perancangan antarmuka *dashboard* pemantauan yang mudah dipahami berhasil diwujudkan, di mana sistem mentransformasikan parameter teknis astronomi yang rumit menjadi representasi grafik interaktif dan indikator warna penilaian risiko (*Traffic Light System*). Melalui langkah visualisasi ini, pengguna awam dapat langsung mengidentifikasi tingkat ancaman objek secara instan. Kemudian, tahapan validasi usability kepada pengguna akhir, yang dieksekusi menggunakan instrumen *System Usability Scale* (SUS) terhadap 30 responden. Hasil evaluasi tersebut meraih skor rata-rata 79,50 (*Acceptable / Grade B*). Rangkaian tahapan ini secara empiris membuktikan bahwa *platform* SINEO berhasil menjawab tantangan untuk menyajikan informasi antariksa yang komunikatif dan berusabilitas tinggi.

6.2. Saran

Berdasarkan batasan masalah yang ditetapkan dalam penelitian ini serta hasil evaluasi akhir sistem SINEO, diajukan beberapa saran rekomendasi teknis untuk pengembangan sistem lebih lanjut di masa mendatang:

1. Perluasan *Scope* Objek Antariksa: Mengingat pembatasan ruang lingkup data objek dekat Bumi dalam penelitian ini hanya mencakup kelompok asteroid saja, pengembangan sistem selanjutnya disarankan untuk memperluas cakupan integrasi data fisis dan mekanika orbit dari kelompok komet agar sistem pemantauan menjadi lebih universal dan komprehensif.
2. Optimalisasi Arsitektur *Hybrid Environment*: Untuk mengoptimalkan ekosistem pengembangan sistem hibrida (*hybrid environment*), disarankan bagi peneliti berikutnya untuk memigrasikan arsitektur komunikasi skrip PHP (Laravel 12) dan *Python engine* yang saat ini menggunakan perintah *shell_exec()* ke dalam protokol komunikasi berbasis *RESTful* API terenkripsi atau memanfaatkan *Message Broker* (seperti *Redis/RabbitMQ*) guna menghindari risiko *I/O bottleneck* dan beban prosesor server lokal yang berlebih.
3. Eksplorasi Variasi Arsitektur *Deep Learning*: Batasan penggunaan model dasar *SimpleRNN* bertumpuk (*Dual-Stacked RNN*) dapat dikembangkan pada

penelitian masa depan dengan menerapkan variasi arsitektur rekuren tingkat lanjut, seperti *Long Short-Term Memory* (LSTM) atau *Gated Recurrent Unit* (GRU), guna membandingkan komparasi ketajaman akurasi serta meminimalkan akumulasi residu *error* pada iterasi proyeksi berantai jangka panjang (*Recursive Forecasting*).

4. Integrasi Kalkulasi Perturbasi Gravitasi: Terkait fokus variabel analisis lintasan dan visualisasi orbit, disarankan untuk mengeksplorasi fungsionalitas sistem agar tidak hanya terpaku pada penyajian analitik tabular, status bahaya *Traffic Light System* (TLS), dan infografis, melainkan turut mengintegrasikan modul perhitungan perturbasi (gangguan gravitasi planet lain) guna mendeteksi pergeseran koordinat orbit secara eksak di alam semesta.
5. Pengembangan Visualisasi 3D Interaktif: Sejalan dengan batasan tampilan pemetaan orbit dua dimensi (2D) sederhana, visualisasi antarmuka detail objek dapat dikembangkan secara interaktif menggunakan pustaka grafis tiga dimensi (3D) berbasis web (seperti *Three.js* atau *WebGL*) agar pengguna dapat melakukan simulasi rotasi ruang angkasa secara spasial dan melipatgandakan fungsi edukasi astronomi publik.
6. Penerapan Audit Pengujian Kode *End-to-End*: Berkenaan dengan pembatasan metode validasi fungsional yang mengandalkan *Black Box Testing* dari sudut pandang masukan dan keluaran antarmuka serta *System Usability Scale* (SUS) untuk usabilitas, proses audit perangkat lunak masa depan perlu dilengkapi dengan metode pengujian internal secara menyeluruh, seperti *White Box Testing* atau *Unit Testing*, untuk menjamin kebersihan dan ketahanan arsitektur baris kode program secara *end-to-end*.