

**PERAMALAN JUMLAH PENGUNJUNG *WEBSITE*
MENGUNAKAN ALGORITMA PROPHET
DENGAN OPTIMASI *SLIME*
*MOULD ALGORITHM***

SKRIPSI

OLEH:

ADELINA BR NAINGGOLAN

228160093



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2026**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

Document Accepted 1/7/26

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Access From (repository.uma.ac.id)1/7/26

**PERAMALAN JUMLAH PENGUNJUNG *WEBSITE*
MENGUNAKAN ALGORITMA PROPHET
DENGAN OPTIMASI *SLIME*
*MOULD ALGORITHM***

SKRIPSI

**Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana di Fakultas Teknik
Universitas Medan Area**

Oleh:

ADELINA BR NAINGGOLAN

228160093

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2026**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

Document Accepted 1/7/26

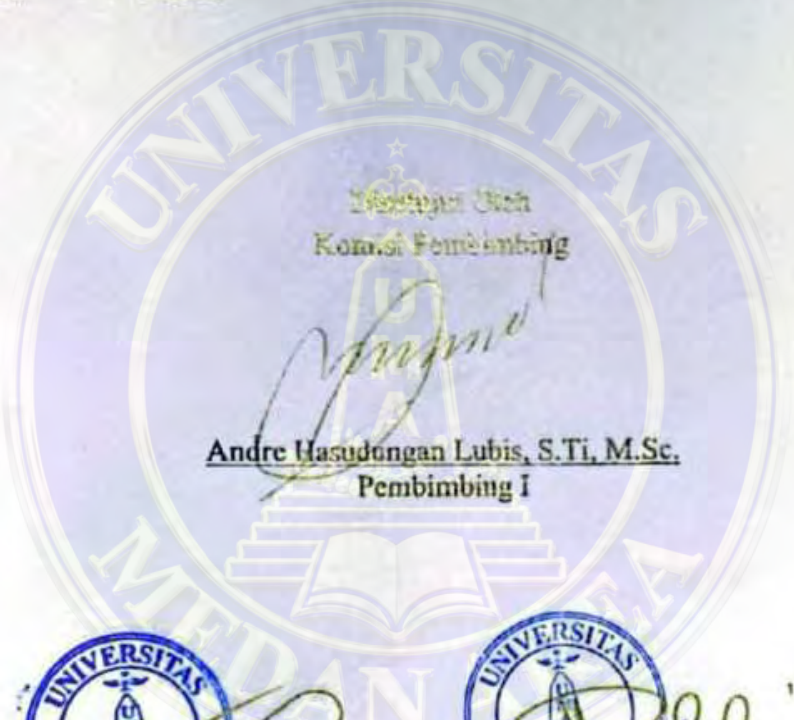
1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

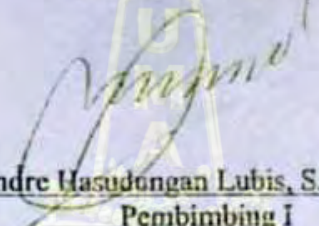
Access From (repository.uma.ac.id)1/7/26

HALAMAN PENGESAHAN

Judul Skripsi : PERAMALAN JUMLAH PENGUNJUNG WEBSITE
MENGGUNAKAN ALGORITMA PROPHET DENGAN
OPTIMASI SLIME MOULD ALGORITHM

Nama : Adelina br Nainggolan
NPM : 220160095
Fakultas : Teknik


Ketua Senat
Komisi Pembimbing


Andre Hasudungan Lubis, S.Ti, M.Sc.
Pembimbing I


Dr. Eng. Supratno, S.T., M.T.
Dekan Fakultas Teknik


Rizki Muliono, S.Kom., M.Kom
Ka. Prodi

Tanggal Lulus: 26 Februari 2026

HALAMAN PERTANYAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 26 Februari 2026



Adelina br Nainggolan
228160093

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : ADELINA BR NAINGGOLAN
NPM : 228160093
Program Studi : Informatika
Fakultas : Teknik
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

Peramalan Jumlah Pengunjung Website Menggunakan Algoritma Prophet dengan Optimasi Slime Mould Algorithm

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan Skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan
Pada tanggal : 26 Februari 2026
Yang menyatakan



(Adelina br Nainggolan)
228160093

RIWAYAT HIDUP

Penulis bernama Adelina br Nainggolan lahir di desa Tanjung |Muda, Kecamatan Air Putih, Kabupaten Batubara pada tanggal 28 Agustus 2004. Penulis merupakan anak ke-5 dari 5 bersaudara dari Bapak Sahat Nainggolan dan Ibu Rosmi Sitorus. Penulis menyelesaikan pendidikan sekolah dasar di SDN. 06 Tanjung Muda dan lulus pada tahun 2016, kemudian melanjutkan ke SMP Negeri 3 Air Putih dan lulus pada tahun 2019. Setelah itu penulis menyelesaikan pendidikan menengah atas di SMK Negeri 1 Air Putih dan lulus pada tahun 2022. Pada tahun 2022 penulis diterima sebagai mahasiswi di Program Studi Informatika, Fakultas Teknik, Universitas Medan Area. Selama menempuh pendidikan, penulis aktif dalam berbagai kegiatan akademik maupun nonakademik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana S.Kom



KATA PENGANTAR

Segala puji dan syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat, hidayah, serta kekuatan yang senantiasa menyertai setiap langkah, sehingga saya dapat menyelesaikan penyusunan skripsi ini yang berjudul **“Peramalan Jumlah Pengunjung Website Menggunakan Algoritma Prophet dengan Optimasi Slime Mould Algorithm”**. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika di Universitas Medan Area.

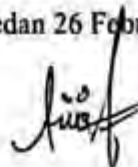
Proses penyusunan skripsi ini bukanlah perjalanan yang mudah. Berbagai tantangan, baik dari segi teknis penelitian, pengolahan data, maupun manajemen waktu, menjadi bagian dari proses yang harus saya hadapi. Namun, berkat doa, dukungan, semangat, serta bimbingan dari berbagai pihak hingga akhirnya skripsi ini dapat diselesaikan dengan baik. Oleh karena itu, dengan kerendahan hati saya ingin menyampaikan terimakasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Dadan Ramdan, M.Eng., M.Sc., selaku Rektor Universitas Medan Area.
2. Bapak Dr.Eng. Supriatno, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Medan Area.
3. Bapak Rizki Muliono, S.Kom., M.Kom, selaku Kepala Program Studi Teknik Informatika Universitas Medan Area.
4. Bapak Andre Hasudungan Lubis, S.Ti, M.Sc., selaku dosen pembimbing yang telah membimbing, mengarahkan, dan memberikan motivasi kepada penulis dalam proses menyelesaikan skripsi ini.

5. Seluruh dosen Program Studi Teknik Informatika Universitas Medan Area yang telah membagikan ilmu, pengalaman, dan wawasan selama masa perkuliahan. Materi, arahan, dan motivasi yang diberikan sangat membantu penulis dalam penyusunan skripsi ini.
6. Terkhusus untuk kedua orang tua tercinta Bapak Sahat Nainggolan dan Ibu Rosmi Sitorus yang selalu memberikan doa, dukungan, kasih sayang, dan semangat tanpa henti kepada penulis. Terima kasih atas pengorbanan dan perhatian yang telah diberikan selama ini. Doa dan dukungan menjadi kekuatan bagi penulis dalam menghadapi setiap tantangan hingga skripsi ini dapat diselesaikan dengan baik.
7. Saudara/saudari tercinta, yaitu Melyanti Nainggolan, Benro Yudis Nainggolan, Handoko Nainggolan, dan Pandi Albinus Nainggolan, yang selalu memberikan dukungan, semangat, dan motivasi selama penulis menyelesaikan skripsi ini.
8. Teman-teman seperjuangan sekaligus sahabat, Vica Sariani Situmorang dan Sisca Oktiani. Terima kasih atas kebersamaan, dukungan, dan momen-momen berharga yang telah kita lalui bersama, perjalanan ini terasa lebih ringan karena kalian. Semoga ikatan ini tetap terjaga meski perkuliahan telah usai.
9. Penulis juga ingin mengucapkan terima kasih kepada diri sendiri yang telah berjuang, bertahan, dan tidak menyerah dalam menghadapi berbagai tantangan selama proses penyusunan skripsi ini. Segala rasa lelah, ragu, dan hambatan yang dilalui menjadi bagian dari perjalanan berharga yang akhirnya dapat terlewati dengan baik.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan dan belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang. Semoga skripsi ini dapat bermanfaat bagi para pembaca dan menjadi referensi untuk penelitian selanjutnya.

Medan 26 Februari 2026



Adelina br Nainggolan
228160093



ABSTRAK

Peramalan deret waktu merupakan permasalahan penting dalam analisis data karena berkaitan dengan kemampuan model dalam menangkap pola tren dan musiman secara akurat. Algoritma Prophet merupakan salah satu metode peramalan deret waktu yang banyak digunakan karena kemampuannya dalam memodelkan komponen tren dan musiman. Namun, kinerja algoritma Prophet sangat dipengaruhi oleh pemilihan *hyperparameter*, sehingga penggunaan parameter *default* sering kali menghasilkan tingkat akurasi yang belum optimal. Oleh karena itu, penelitian ini bertujuan untuk meningkatkan kinerja model Prophet melalui optimasi *hyperparameter* menggunakan *Slime Mould Algorithm* (SMA). Penelitian ini menggunakan data deret waktu sebagai studi kasus pengujian model. Tahapan penelitian meliputi prapemrosesan data, proses optimasi *hyperparameter* Prophet menggunakan SMA, pembangunan model peramalan, serta evaluasi kinerja model. *Hyperparameter* yang dioptimasi meliputi *changepoint prior scale* dan *seasonality prior scale*. Evaluasi kinerja model dilakukan dengan membandingkan nilai kesalahan menggunakan metrik *Root Mean Squared Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE). Hasil penelitian menunjukkan bahwa optimasi *hyperparameter* menggunakan SMA mampu meningkatkan kinerja model Prophet. Nilai RMSE berhasil diturunkan dari 399,7137 menjadi 338,0901, sedangkan nilai MAPE menurun dari 13,5% menjadi 11,47%. Selain itu, hasil peramalan menunjukkan bahwa model Prophet yang telah dioptimasi lebih adaptif terhadap perubahan tren dan fluktuasi pada data deret waktu. Dengan demikian, SMA terbukti efektif sebagai metode optimasi *hyperparameter* dalam meningkatkan akurasi dan stabilitas model peramalan Prophet.

Kata Kunci: Peramalan deret waktu, Prophet, *Slime Mould Algorithm*, Optimasi *hyperparameter*, *Metaheuristik*

ABSTRACT

Time series forecasting is a crucial issue in data analysis because it is related to the model's ability to accurately capture trend and seasonal patterns. The Prophet algorithm is a widely used time series forecasting method due to its ability to model trend and seasonal components. However, the performance of the Prophet algorithm is strongly influenced by hyperparameter selection, so the use of default parameters often results in a suboptimal level of accuracy. Therefore, this study aims to improve the performance of the Prophet model through hyperparameter optimization using the Slime Mold Algorithm (SMA). This study uses time series data as a case study for model testing. The research stages include data preprocessing, the Prophet hyperparameter optimization process using SMA, forecasting model development, and model performance evaluation. The optimized hyperparameters include the changepoint prior scale and the seasonality prior scale. Model performance evaluation is carried out by comparing error values using the Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) metrics. The results show that hyperparameter optimization using SMA can improve the performance of the Prophet model. The RMSE value was successfully reduced from 399.7137 to 338.0901, while the MAPE value decreased from 13.5% to 11.47%. Furthermore, the forecasting results show that the optimized Prophet model is more adaptive to changing trends and fluctuations in time series data. Thus, the Slime Mold Algorithm has proven effective as a hyperparameter optimization method in improving the accuracy and stability of the Prophet forecasting model.

Keywords: Time series forecasting, Prophet, Slime Mold Algorithm, Hyperparameter optimization, Metaheuristics

DAFTAR ISI

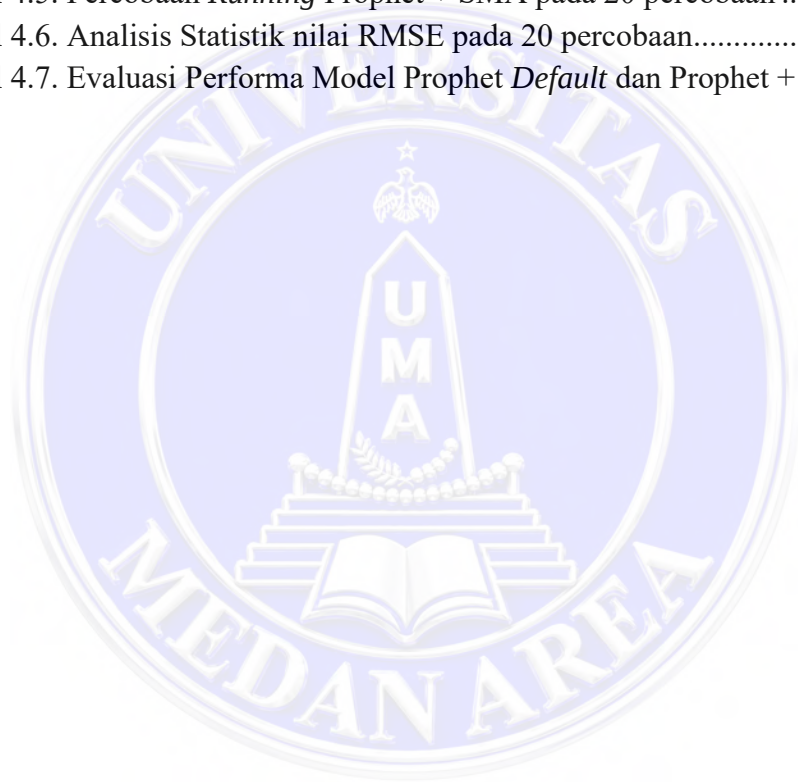
	Halaman
HALAMAN PENGESAHAN.....	i
HALAMAN PERTANYAAN	ii
RIWAYAT HIDUP	iv
KATA PENGANTAR.....	v
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA	6
2.1. Peramalan	6
2.1.1. Evaluasi Model Peramalan.....	6
2.2. Algoritma Prophet	8
2.2.1. Alur Kerja Prophet.....	9
2.2.2. Flowchart Algoritma Prophet.....	11
2.3. <i>Hyperparameter</i>	13
2.4. SMA	14
2.4.1. Alur Kerja SMA.....	16
2.4.2. <i>Flowchart</i> SMA	17
2.5. Penelitian Terdahulu.....	19
BAB III METODOLOGI PENELITIAN	20
3.1. Tahapan Penelitian	20
3.2. Pengumpulan Data	22
3.3. Prapemrosesan Data	22
3.4. Optimasi <i>Hyperparameter</i>	23
3.5. Implementasi Prophet.....	25
3.6. Evaluasi Model.....	26
BAB IV HASIL DAN PEMBAHASAN	28
4.1. Karakteristik Data.....	28
4.2. Hasil Performa Algoritma Prophet <i>Default</i>	30
4.3. Hasil Prophet + SMA	34

4.4.	Analisis Statistik.....	37
4.4.1.	Percobaan <i>Running</i> Prophet <i>Default</i>	37
4.4.2.	Percobaan <i>Running</i> Prophet + SMA.....	38
4.4.3.	Analisis Statistik nilai RMSE	40
4.5.	Uji konvergensi	41
4.6.	Pembahasan	43
BAB V KESIMPULAN DAN SARAN.....		46
5.1.	Kesimpulan.....	46
5.2.	Saran.....	47
DAFTAR PUSTAKA.....		48
LAMPIRAN.....		53



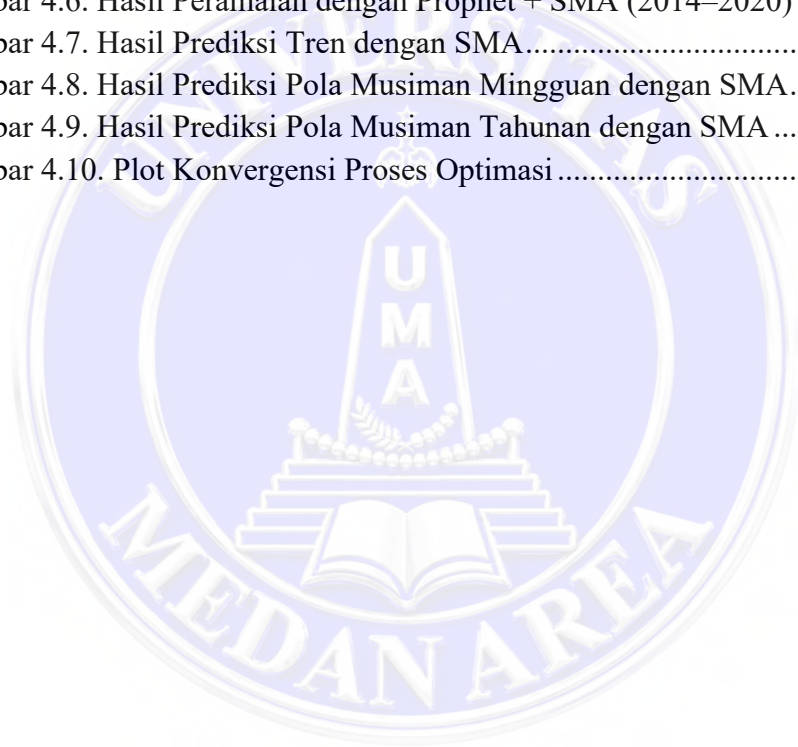
DAFTAR TABEL

	Halaman
Tabel 2.1. Penelitian Terdahulu	19
Tabel 3.1. <i>Dataset</i> Jumlah Pengunjung <i>Website</i>	22
Tabel 3.2. Rentang Pencarian.....	24
Tabel 4.1. Statistik Deskriptif Data <i>First Time Visits</i> (2014 – 2020).....	28
Tabel 4.2. Hasil Evaluasi Model Prophet <i>Default</i>	30
Tabel 4.3. Hasil Evaluasi Model Prophet + SMA.....	34
Tabel 4.4. Percobaan <i>Running</i> Prophet <i>Default</i> pada 20 percobaan	38
Tabel 4.5. Percobaan <i>Running</i> Prophet + SMA pada 20 percobaan	39
Tabel 4.6. Analisis Statistik nilai RMSE pada 20 percobaan.....	40
Tabel 4.7. Evaluasi Performa Model Prophet <i>Default</i> dan Prophet + SMA	43



DAFTAR GAMBAR

	Halaman
Gambar 2.1. <i>Flowchart</i> Algoritma Prophet	12
Gambar 2.2. <i>Flowchart</i> SMA.....	18
Gambar 3.1. <i>Flowchart</i> Tahapan Penelitian.....	20
Gambar 4.1. Rata-rata Jumlah Pegunjung <i>Website</i> Tahunan (2014–2020)	29
Gambar 4.2. Hasil Peramalan dengan Algoritma Prophet (2014–2020).....	31
Gambar 4.3. Hasil Prediksi Tren dengan Prophet	32
Gambar 4.4. Hasil Prediksi Pola Musiman dengan Prophet	32
Gambar 4.5. Hasil Prediksi Pola Musiman Tahunan dengan Prophet	33
Gambar 4.6. Hasil Peramalan dengan Prophet + SMA (2014–2020)	34
Gambar 4.7. Hasil Prediksi Tren dengan SMA.....	35
Gambar 4.8. Hasil Prediksi Pola Musiman Mingguan dengan SMA.....	36
Gambar 4.9. Hasil Prediksi Pola Musiman Tahunan dengan SMA	36
Gambar 4.10. Plot Konvergensi Proses Optimasi	42



DAFTAR LAMPIRAN

	Halaman
1. Kode Program	53



BAB I

P ENDAHULUAN

1.1. Latar Belakang

Peramalan deret waktu merupakan salah satu topik penting dalam kecerdasan buatan dan analisis data karena membantu meramalkan pola data yang berubah-ubah dan tidak tetap (Naufal, 2024). Banyak algoritma telah diciptakan untuk meningkatkan ketepatan peramalan, tetapi masalah besar yang sering muncul adalah kemampuan model untuk mengenali perubahan dalam tren, pola musiman, dan variasi data yang rumit (Rozy dkk., 2023). Selain itu, kinerja suatu algoritma peramalan sangat dipengaruhi oleh pengaturan parameter model yang digunakan (Hakimah dkk., 2020).

Salah satu metode yang cukup populer digunakan dalam analisis data deret waktu adalah algoritma Prophet, yang dikembangkan oleh Facebook (Meta) (Hidayat & Sulistiyono, 2025). Prophet merupakan metode berbasis model aditif yang menggabungkan tiga komponen utama yaitu tren, musiman, dan efek liburan (Anamsyah dkk., 2025). Keunggulan utama Prophet terletak pada kemampuannya menangani data dengan pola yang kompleks, musiman yang tidak teratur, serta ketahanannya terhadap data hilang (*missing value*). Selain itu, Prophet relatif mudah digunakan karena dapat menyesuaikan tren dan pola data secara otomatis, sehingga banyak diterapkan untuk berbagai bidang seperti peramalan penjualan, jumlah pengunjung *website*, hingga permintaan energi (Santoso & Dewi, 2024).

Namun, dibalik keunggulannya, algoritma Prophet juga memiliki beberapa kelemahan. Salah satunya adalah sensitivitas terhadap pemilihan *hyperparameter* awal seperti *changepoint prior scale* dan *seasonality prior scale*, yang dapat mempengaruhi hasil peramalan secara signifikan (Atamimi dkk., 2025). Kesalahan dalam menentukan nilai *hyperparameter* ini dapat menyebabkan model menjadi *overfitting* atau *underfitting*, sehingga hasil prediksi menjadi kurang akurat (Manandhar dkk., 2024).

Untuk mengatasi masalah tersebut, dibutuhkan suatu mekanisme optimasi *hyperparameter* agar nilai-nilai parameter pada Prophet dapat disesuaikan secara optimal sesuai karakteristik data. Optimasi *hyperparameter* bertujuan untuk meningkatkan performa model dengan mencari kombinasi parameter terbaik yang menghasilkan tingkat kesalahan (*error*) paling rendah (Rusman dkk., 2023). Salah satu pendekatan yang dapat digunakan untuk proses ini adalah dengan algoritma *metaheuristik*, salah satunya adalah *Slime Mould Algorithm* (SMA). Algoritma ini terinspirasi dari perilaku organisme lendir (*Physarum polycephalum*) dalam mencari makanan melalui mekanisme adaptif dan efisien (Tang dkk., 2021). SMA dikenal memiliki kemampuan eksplorasi dan eksploitasi ruang solusi yang seimbang serta mampu menghindari jebakan pada solusi lokal (Duan dkk., 2025). Dengan menggabungkan Prophet dan SMA, diharapkan proses tuning *hyperparameter* Prophet dapat berjalan otomatis dan lebih optimal, sehingga meningkatkan akurasi hasil peramalan (Alamsyah dkk., 2025).

Beberapa penelitian sebelumnya menunjukkan bahwa algoritma Prophet efektif digunakan untuk peramalan data deret waktu. Penelitian oleh (Simamora dkk., 2024) menunjukkan bahwa algoritma Prophet efektif untuk memprediksi

harga saham PT United Tractors Tbk dengan tingkat akurasi tinggi dan (MAPE) sebesar 3,86%. Namun, penelitian tersebut belum membahas secara khusus pengaruh pengaturan *hyperparameter* terhadap peningkatan performa model Prophet. Selain itu, pada penelitian (Atamimi dkk., 2025) menerapkan tuning *hyperparameter* Prophet pada data penjualan ritel yang *sparse* dan berhasil meningkatkan akurasi prediksi, menunjukkan bahwa optimasi *hyperparameter* dapat menurunkan kesalahan peramalan. Berdasarkan kedua penelitian tersebut, Prophet terbukti efektif untuk peramalan deret waktu, dan penerapan optimasi menggunakan algoritma seperti SMA berpotensi meningkatkan performanya.

Berdasarkan uraian tersebut, penelitian ini mengusulkan penerapan SMA sebagai metode optimasi *hyperparameter* pada algoritma Prophet. Pendekatan ini diharapkan mampu meningkatkan performa model Prophet dalam melakukan peramalan dengan tingkat kesalahan yang lebih rendah. Fokus penelitian ini adalah menganalisis pengaruh proses optimasi *hyperparameter* menggunakan SMA terhadap kinerja algoritma Prophet dalam memodelkan data deret waktu, sehingga dapat menjadi acuan dalam pengembangan metode peramalan berbasis optimasi *metaheuristik* pada penelitian selanjutnya.

1.2. Rumusan Masalah

Bagaimana meningkatkan performa model peramalan berbasis algoritma Prophet melalui optimasi *hyperparameter* menggunakan SMA agar mampu menghasilkan model yang lebih adaptif terhadap pola data deret waktu yang kompleks serta memiliki tingkat kesalahan prediksi yang lebih rendah?.

1.3. Batasan Masalah

1. Penelitian ini berfokus pada peningkatan performa model peramalan berbasis algoritma Prophet melalui optimasi *hyperparameter* dalam memodelkan data deret waktu, dengan data jumlah pengunjung *website* digunakan sebagai studi kasus.
2. Proses optimasi hanya mencakup *hyperparameter* utama Prophet seperti *changepoint prior scale* dan *seasonality prior scale*.
3. Dataset yang digunakan merupakan data sekunder yang diakses melalui tautan <https://www.kaggle.com/datasets/bobnau/daily-website-visitors/code>
4. Dataset terdiri dari Tanggal dan Jumlah Pengunjung
5. Penelitian ini tidak membahas aspek di luar pemodelan dan evaluasi algoritma dan hanya berfokus pada kinerja model peramalan.
6. Penelitian ini tidak melakukan perbandingan dengan algoritma *metaheuristik* lainnya.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah untuk meningkatkan performa algoritma Prophet melalui optimasi *hyperparameter* menggunakan SMA, dalam meramalkan data jumlah pengunjung *website* sebagai studi kasus.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Secara teoritis penelitian ini dapat menambah pemahaman tentang bagaimana algoritma Prophet bekerja dalam memodelkan data deret waktu, serta menunjukkan bahwa performa Prophet dapat ditingkatkan melalui proses optimasi *hyperparameter*.
2. Secara praktis, hasil penelitian ini menghasilkan model Prophet yang telah dioptimasi sehingga dapat menjadi contoh atau referensi dalam membangun model peramalan yang lebih akurat dan stabil pada data deret waktu.
3. Secara metodologis, penelitian ini Memberikan gambaran penerapan pendekatan gabungan antara Prophet dan SMA sebagai metode optimasi yang dapat dijadikan acuan dalam penelitian-penelitian serupa di masa mendatang

BAB II

TINJAUAN PUSTAKA

2.1. Peramalan

Peramalan (*forecasting*) merupakan suatu proses memperkirakan nilai atau kejadian di masa depan berdasarkan data *historis* dan pola yang terjadi sebelumnya (Syaliman dkk., 2023). Peramalan dapat didefinisikan sebagai seni dan ilmu memprediksi kejadian di masa depan dengan menggunakan data masa lalu yang dianalisis untuk menemukan pola atau hubungan tertentu (Nugraha, 2025). Proses peramalan menjadi hal penting dalam berbagai bidang, termasuk ekonomi, cuaca, kesehatan, dan teknologi informasi.

Dalam bidang teknologi informasi, peramalan sering dimanfaatkan untuk berbagai hal, misalnya memprediksi jumlah pengunjung *website*, beban pada server, lalu lintas jaringan, dan juga pola perilaku pengguna. Peramalan yang tepat dapat membantu pengambilan keputusan dalam merencanakan sumber daya yang dibutuhkan, strategi pemasaran, serta perbaikan layanan (Yao dkk., 2023). Dengan demikian, pemilihan metode peramalan yang tepat serta penyesuaian *hyperparameter* model menjadi aspek penting untuk mencapai hasil prediksi yang terbaik (Madhusudhanan dkk., 2024).

2.1.1. Evaluasi Model Peramalan

Evaluasi hasil peramalan sangat penting dilakukan untuk mengetahui sejauh mana model mampu memberikan prediksi yang akurat. Terdapat beberapa metrik

evaluasi yang umum digunakan dalam penelitian peramalan antara lain sebagai berikut:

1. *Root Mean Squared Error (RMSE)*

RMSE adalah metrik yang diperoleh dari akar kuadrat rata-rata selisih kuadrat antara nilai prediksi dan nilai aktual. RMSE lebih peka terhadap *outlier* karena setiap kesalahan dikuadratkan sebelum dihitung rata-ratanya, sehingga kesalahan yang besar akan memiliki pengaruh lebih signifikan (Hodson, 2022). Oleh karena itu, RMSE sangat berguna dalam situasi di mana kesalahan dengan nilai besar perlu diminimalkan atau dianggap sangat penting. Berikut adalah rumus perhitungan RMSE beserta penjelasan dari masing-masing komponennya yang mengacu pada (Karunasingha, 2022).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.1)$$

n : jumlah total data pengamatan (observasi).

y_i : nilai aktual (nilai sebenarnya) pada data ke- i .

$\hat{y}_i$: nilai hasil prediksi model pada data ke- i .

$(y_i - \hat{y}_i)^2$: kuadrat selisih antara nilai aktual dan hasil peramalan

$\sqrt{}$: akar kuadrat dari nilai rata-rata kesalahan

RMSE sering digunakan untuk menilai stabilitas model dalam menghadapi variasi data yang ekstrim.

2. Mean Absolute Percentage Error (MAPE)

MAPE adalah cara untuk mengukur seberapa besar kesalahan hasil prediksi dalam bentuk persen dari nilai sebenarnya (Kurniawan dkk., 2025). MAPE memberikan gambaran untuk mengetahui seberapa jauh hasil prediksi model berbeda dari data aslinya, sehingga lebih mudah memahami tingkat akurasi model dalam ukuran yang sederhana dan jelas. Berikut adalah rumus perhitungan MAPE .

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.2)$$

RMSE dan MAPE adalah metrik yang digunakan untuk menilai kinerja model peramalan. RMSE menunjukkan besar kesalahan prediksi dalam satuan yang sama dengan data asli dan lebih sensitif terhadap kesalahan yang besar. Sementara itu, MAPE menunjukkan kesalahan prediksi dalam bentuk persentase sehingga lebih mudah dipahami dan memudahkan perbandingan antar model. Oleh karena itu, penggunaan RMSE dan MAPE secara bersama-sama dapat memberikan gambaran yang lebih jelas tentang akurasi model peramalan (Amansyah dkk., 2024).

2.2. Algoritma Prophet

Algoritma Prophet dikembangkan oleh Facebook sebagai alat prediksi yang mampu melakukan peramalan data deret waktu (*time series*) secara cepat dan efisien (Wiejaya & Fenriana, 2024). Model ini menerapkan pendekatan *additive regression* model dengan tren non-linear dalam proses peramalannya . Algoritma Prophet terdiri dari beberapa komponen utama, antara lain komponen tren yang menggambarkan perubahan jangka panjang pada data, komponen musiman yang

menangkap pola berulang, serta komponen hari libur yang mempertimbangkan pengaruh tanggal-tanggal khusus (Taofiqurrohman dkk., 2025).

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (2.4)$$

Keterangan:

$y(t)$ = nilai hasil peramalan (prediksi) pada waktu t

$g(t)$ = fungsi tren, yang memodelkan perubahan jangka panjang (bisa *linier* atau *logistik*)

$s(t)$ = fungsi *seasonality*, yang merepresentasikan pola berulang seperti harian, mingguan, atau tahunan

$h(t)$ = fungsi *holidays*, efek hari libur atau kejadian khusus yang memengaruhi data

ε_t = *error term*, yaitu komponen acak yang tidak dapat dijelaskan oleh model.

2.2.1. Alur Kerja Prophet

Secara umum, alur kerja Prophet dalam melakukan peramalan jumlah pengunjung *website* adalah sebagai berikut:

1. Menyiapkan Data Kunjungan Website

Mulai dengan menyiapkan data kunjungan *website*. Data disusun dalam dua kolom: tanggal kunjungan (ds) dan jumlah pengunjung (y).

2. Prapemrosesan Data

Pada tahap ini dilakukan pemeriksaan terhadap *missing value*, duplikasi data, dan anomali seperti lonjakan atau penurunan ekstrim. Data kemudian diperbaiki dan diseragamkan, termasuk memastikan format tanggal sesuai agar Prophet dapat memprosesnya.

3. Pembangunan Model Prophet

Setelah data siap, model Prophet dibuat dengan dua komponen utama yaitu *trend* yang menunjukkan pola naik-turun pengunjung, dan *seasonality* yang menangkap pola berulang harian, mingguan, atau bulanan.

4. Pengaturan *Hyperparameter* Model

Pengaturan *Hyperparameter* penting dilakukan untuk melihat sensitivitas terhadap perubahan tren (*changepoint_prior_scale*) dan seberapa fleksibel model mengikuti pola musiman (*seasonality_prior_scale*).

5. Pelatihan Model (Model *Fitting*)

Prophet dilatih menggunakan data *historis* untuk mempelajari tren jangka panjang, musiman, dan titik perubahan. Hasilnya model mampu merepresentasikan pola kunjungan *website*.

6. Proses Peramalan (*Forcaseting*)

Setelah model terlatih, Prophet digunakan untuk menghasilkan prediksi jumlah pengunjung *website* pada masa mendatang. Selain menghasilkan nilai prediksi, Prophet juga memberikan perkiraan rentang kemungkinan jumlah pengunjung sebenarnya.

7. Evaluasi Model

Langkah terakhir, dilakukan evaluasi untuk mengetahui seberapa akurat prediksi yang dihasilkan dengan membandingkan hasil prediksi dengan data sebenarnya menggunakan metrik seperti RMSE atau MAPE. Tahap ini memastikan bahwa model Prophet cukup akurat dan dapat digunakan sebagai dasar pengambilan keputusan dalam pengelolaan *website*.

Berikut adalah *pseudocode* dari algoritma Prophet:

Algoritma Prophet

Input:

Dataset $D = \{(ds, s)\}$

Parameters: cp, s

Output:

Forecast $\hat{y}(t)$, RMSE

Procedure Prophet_Forecasting(D):

1. Load dataset D (ds = tanggal, y = jumlah pengunjung)
2. Preprocess data:
 - ubah ds ke *datetime*
 - isi *missing value* bila ada
3. Split data $\rightarrow$ train (80%), test (20%)
4. Inisialisasi model Prophet:

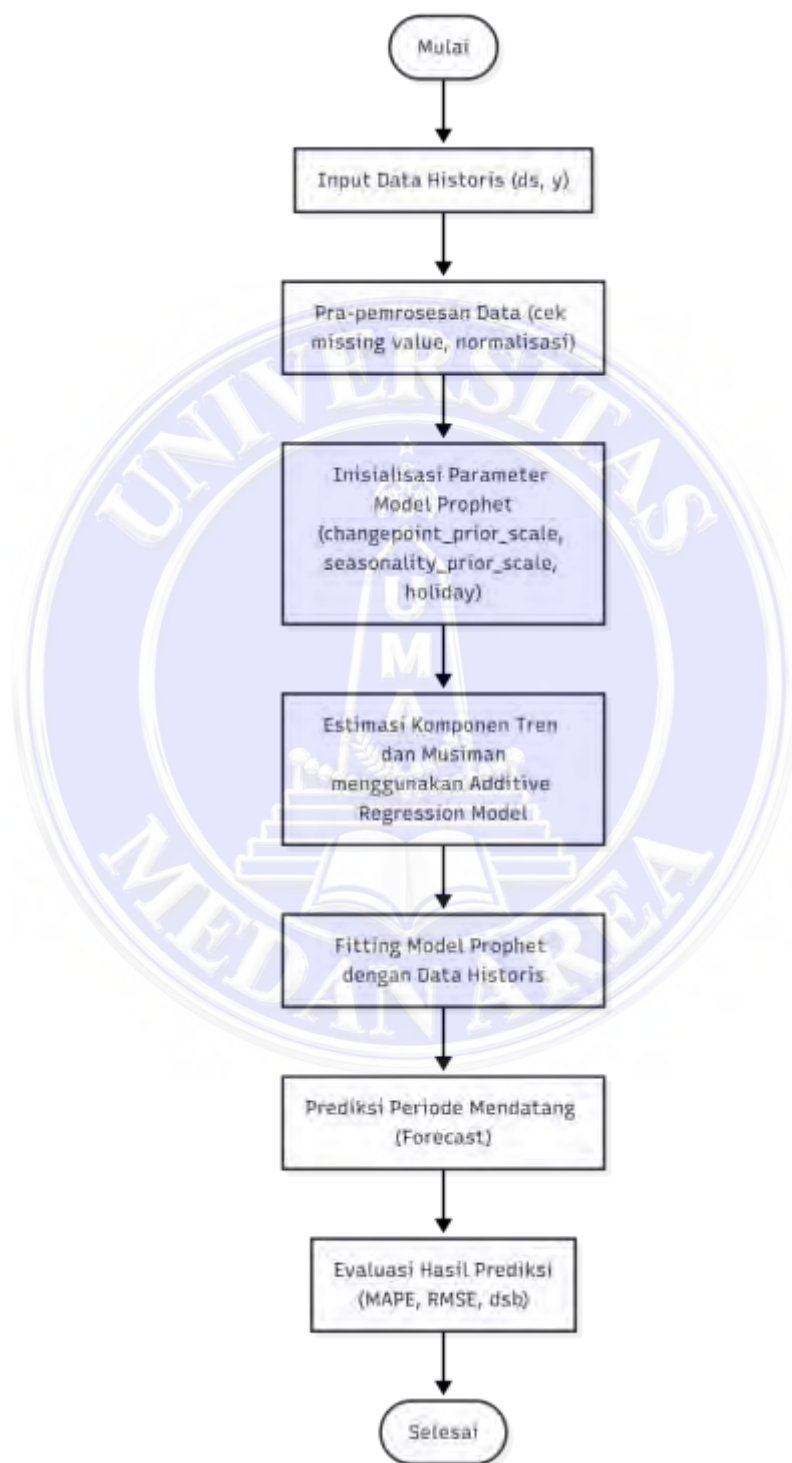

```
model  $\leftarrow$  Prophet(
    changepoint_prior_scale = cp,
    seasonality_prior_scale = s,
    yearly_seasonality = True,
    weekly_seasonality = True
)
```
5. Fit model dengan data train
6. Buat future dataframe sepanjang $\text{len}(\text{test})$
7. Lakukan prediksi $\rightarrow \text{forecast} \leftarrow \text{model.predict}(\text{future})$
8. Ambil nilai $y_{pred} \leftarrow \text{forecast}['\hat{y}'][-\text{len}(\text{test}):]$
9. Hitung $\text{RMSE} = \sqrt{\text{mean}((y_{test} - y_{pred})^2)}$
10. Visualisasikan hasil (aktual vs prediksi)
11. Return y_{pred} , RMSE

End Procedure

2.2.2. Flowchart Algoritma Prophet

Flowchart adalah diagram alur yang digunakan untuk menggambarkan urutan Langkah atau proses dalam suatu sistem, algoritma, atau program dengan cara yang terstruktur dan jelas (Smrti dkk., 2023). Setiap langkah dalam diagram alur diwakili oleh simbol tertentu, seperti persegi panjang untuk proses, belah ketupat untuk menunjukkan keputusan, dan panah untuk menggambarkan arah dari alur tersebut. Melalui *flowchart* ini, pengguna dapat memahami alur kerja Prophet mulai dari input data *historis*, pembersihan dan persiapan data, penentuan *hyparameter* utama seperti *changepoint prior scale* dan *seasonality prior scale*,

hingga tahap *fitting* model, pembuatan peramalan, dan evaluasi hasil prediksi menggunakan metrik kesalahan seperti MAPE atau RMSE (Anwar & Rassiyanti, 2025). Berikut adalah *flowchart* Algoritma Prophet :



Gambar 2.1. *Flowchart* Algoritma Prophet

2.3. *Hyperparameter*

Dalam hal peramalan menggunakan algoritma Prophet, *hyperparameter* mengatur seberapa fleksibel model dalam menangkap pola tren dan musiman pada data deret waktu (Hidayat dkk., 2025). Memilih nilai *hyperparameter* yang tepat akan menghasilkan model yang akurat, sementara pilihan yang salah dapat menyebabkan model terlalu mengikuti data latih (*overfitting*) atau tidak mampu menangkap pola data dengan baik (*underfitting*) (Dinata & Hasdyna, 2025). Berikut adalah beberapa *hyperparameter* utama Prophet:

1. *Changepoint_Prior_Scale*

Changepoint_Prior_Scale menentukan sejauh mana model dapat beradaptasi dengan perubahan tren dalam data deret waktu. Nilai yang besar membuat model lebih sensitif terhadap perubahan tren kecil, sedangkan nilai kecil menghasilkan tren yang lebih halus. Pemilihan nilai ini penting untuk menyeimbangkan antara kemampuan model dalam mengikuti perubahan data dan menghindari *overfitting* (Atamimi dkk., 2025).

2. *Seasonality_prior_scale*

seasonality_prior_scale mengatur sejauh mana model dapat menyesuaikan dengan pola musiman yang ada dalam data. Nilai yang besar memungkinkan Prophet untuk mendeteksi variasi musiman yang rumit, sedangkan nilai kecil mengarah pada pola musiman yang lebih sederhana dan stabil. *hyperparameter* ini bermanfaat ketika data menunjukkan variasi yang teratur, seperti pola kunjungan setiap minggu atau setiap bulan (Dai dkk., 2025).

2.4. SMA

SMA adalah algoritma metaheuristik yang terinspirasi dari perilaku jamur lendir dalam mencari makanan (Qiu dkk., 2022). Organisme ini mampu mendeteksi sumber makanan di sekitarnya dan membentuk jalur menuju sumber tersebut. Perilaku mencari makanan jamur lendir dibagi menjadi tiga kategori yaitu menemukan sumber makanan, mendekati sumber makanan, dan mencerna makanan menggunakan enzim. Selama proses migrasi, bagian depan jamur lendir berkembang menjadi bentuk menyerupai kipas dan menciptakan jaringan pembuluh darah dengan variasi ketebalan pada berbagai sumber makanan (Wei dkk., 2024).

Algoritma ini dapat dimanfaatkan untuk menyelesaikan masalah optimasi atau perencanaan jalur karena kemampuan adaptasi dan penyebarannya yang luar biasa. Keunikan kemampuan ini menjadikan *Slime Mould* sebagai inspirasi penting dalam optimasi algoritma pengembangan yang efektif (Gharehchopogh dkk., 2023). SMA terdiri dari tiga langkah utama, yaitu inisialisasi populasi, pembaruan posisi individu berdasarkan nilai *fitness*, dan seleksi solusi terbaik. Langkah-langkah ini memungkinkan algoritma untuk meniru cara *slime mould* beradaptasi dengan lingkungannya dan bergerak menuju titik optimal.

1. Inisialisasi Populasi

Setiap solusi awal (*slime mould*) diinisialisasi secara acak dalam ruang pencarian menggunakan rumus:

$$X_i = X_{min} + r \cdot (X_{max} - X_{min}) \quad (2.5)$$

Dengan:

X_i = posisi individu ke- i ,

$(X_{max} - X_{min})$ =batas bawah dan atas variabel,

r = bilangan acak uniform dalam rentang $[0, 1]$

2. Pembaruan Bobot Berdasarkan Nilai *Fitness*

Bobot tiap individu menggambarkan intensitas bau makanan yang mengarahkan pencarian menuju solusi terbaik. Bobot dihitung menggunakan persamaan:

$$W_i = \begin{cases} 1 + \log \left(\frac{best - f(X_i)}{best - worst + \varepsilon} \right), & \text{jika } f(X_i) < best \\ 1 - \log \left(\frac{f(X_i) - worst}{best - worst + \varepsilon} \right), & \text{lainnya} \end{cases} \quad (2.6)$$

Dengan:

$f(X_i)$ = nilai *fitness* individu ke-i,

$best - worst$ = nilai *fitness* terbaik dan terburuk dalam populasi,

ε = konstanta kecil untuk mencegah pembagian nol.

3. Pembaruan Posisi *Slime Mould*

Posisi setiap individu diperbarui dengan mempertimbangkan pengaruh solusi terbaik serta dua individu acak lainnya menggunakan rumus:

$$X_i^{(t+1)} = \begin{cases} X_i^{(t+1)} + vb. (W_i \cdot X_A^{(t)} - X_B^{(t)}), & \text{jika } r < p \\ vb. X_i^{(t)}, & \text{jika } r > p \end{cases} \quad (2.7)$$

Dengan:

$X(t)_{best}$ = posisi terbaik saat iterasi ke-t,

X_A, X_B = dua posisi acak dalam populasi,

vb = faktor pembobot acak dalam rentang $[-a, a]$,

p = probabilitas adaptif yang tergantung pada nilai *fitness*.

Nilai a dihitung dengan:

$$a = \tanh \left(\left| \frac{iter}{\max_iter} - 1 \right| \right) \quad (2.8)$$

4. Seleksi Solusi Terbaik

Setelah pembaruan posisi, setiap individu dievaluasi kembali berdasarkan fungsi tujuan (objective function). Solusi dengan nilai fitness terbaik disimpan sebagai Xbest dan akan digunakan untuk iterasi berikutnya hingga mencapai kondisi konvergensi atau batas iterasi maksimum.

2.4.1. Alur Kerja SMA

Berikut langkah-langkah umum dari alur kerja SMA:

1. Inisialisasi populasi

Membentuk populasi awal *slime mould* acak di ruang pencarian (*search space*). Contoh: setiap individu mewakili kombinasi nilai *hyperparameter* Prophet seperti *changepoint_prior_scale* dan *seasonality_prior_scale*.

2. Evaluasi fitness

Menghitung *fitness* setiap individu berdasarkan fungsi objektif, misalnya minimasi *error* (MAPE) hasil peramalan Prophet.

3. Perangkingan solusi

Mengurutkan *slime mould* berdasarkan nilai *fitness* terbaik hingga terburuk.

4. Pembaruan bobot (*Weight updating*)

Menghitung bobot berdasarkan tingkat *concentration of food* (semakin baik solusi → semakin besar bobot).

5. Pembaruan posisi (*Position updating*)

Setiap *slime mould* memperbarui posisinya menggunakan persamaan adaptif yang mempertimbangkan bobot, posisi terbaik saat ini, dan faktor osmotik.

6. Pemeriksaan batasan (*Boundary Control*)

Memastikan semua solusi tetap dalam batas ruang pencarian yang *valid* (tidak keluar dari nilai *hyperparameter* yang ditentukan).

7. Seleksi solusi terbaik

Menyimpan solusi terbaik sejauh iterasi berlangsung.

8. Kriteria berhenti

Proses iteratif berhenti ketika mencapai jumlah iterasi maksimum atau konvergensi solusi tercapai.

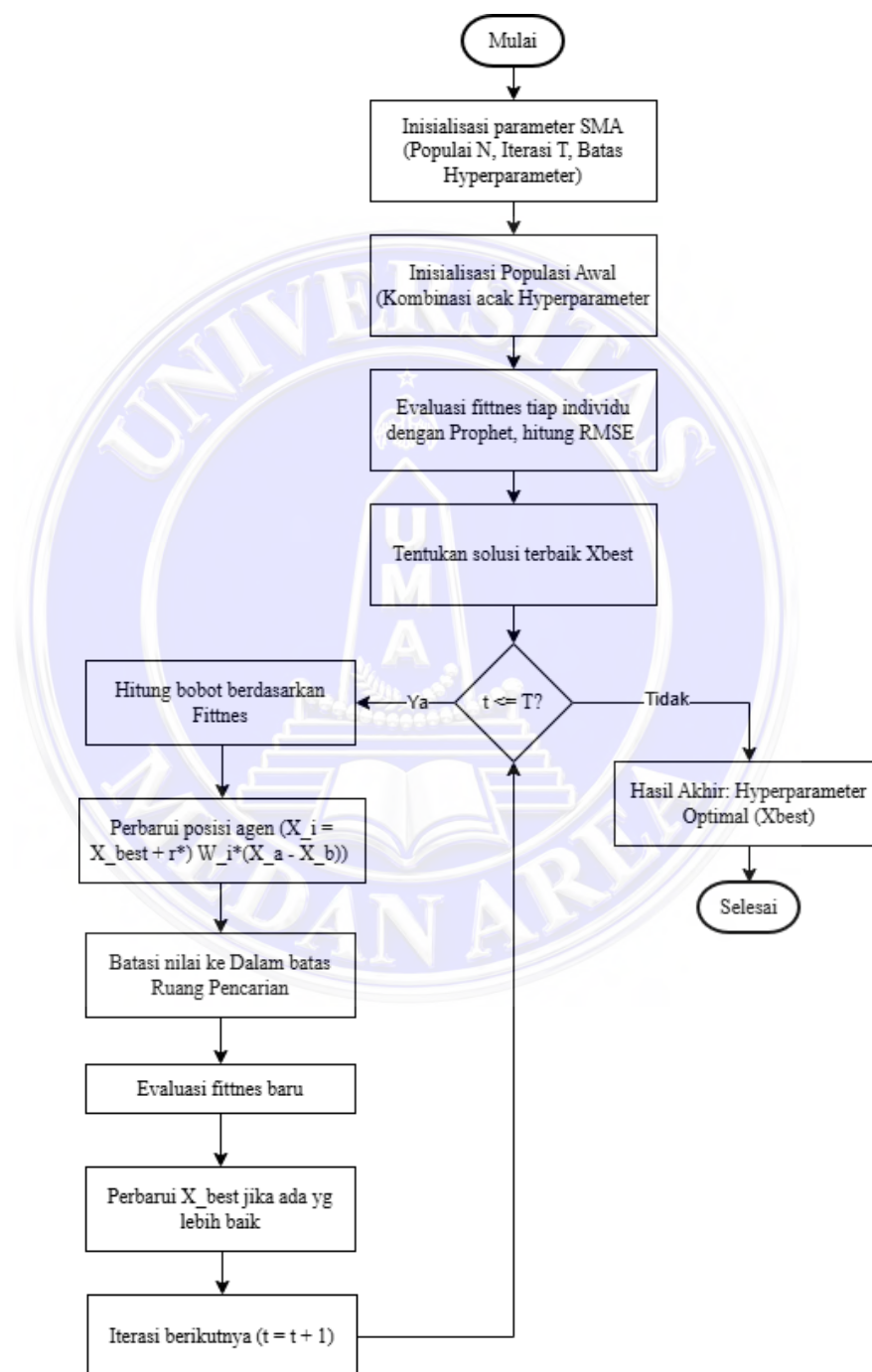
9. Hasil optimasi

Nilai *hyperparameter* terbaik (hasil SMA) digunakan untuk melatih model Prophet agar menghasilkan prediksi paling akurat.

2.4.2. Flowchart SMA

Flowchart SMA adalah diagram alur yang digunakan untuk menjelaskan langkah-langkah kerja algoritma SMA secara berurutan dan jelas (Fauzi, 2020). Dalam *flowchart* ini ditunjukkan bagaimana proses dimulai dari membuat populasi awal secara acak, lalu menghitung nilai kemampuan (*fitness*) tiap individu berdasarkan hasil prediksi model Prophet. Setelah itu, algoritma mencari posisi terbaik dan memperbarui posisi semua individu dengan meniru cara jamur lendir bergerak melalui tekanan osmotik. Proses ini dilakukan berulang-ulang sampai mencapai hasil yang stabil atau jumlah percobaan yang ditentukan. Melalui

flowchart ini, kita bisa lebih mudah memahami bagaimana SMA membantu menemukan nilai *hyperparameter* terbaik agar model Prophet menghasilkan peramalan jumlah pengunjung *website* yang paling akurat (Wei dkk., 2024). berikut adalah gambar *Flowchart* SMA:



Gambar 2.2. *Flowchart* SMA

2.5. Penelitian Terdahulu

Tabel 2.1. Penelitian Terdahulu

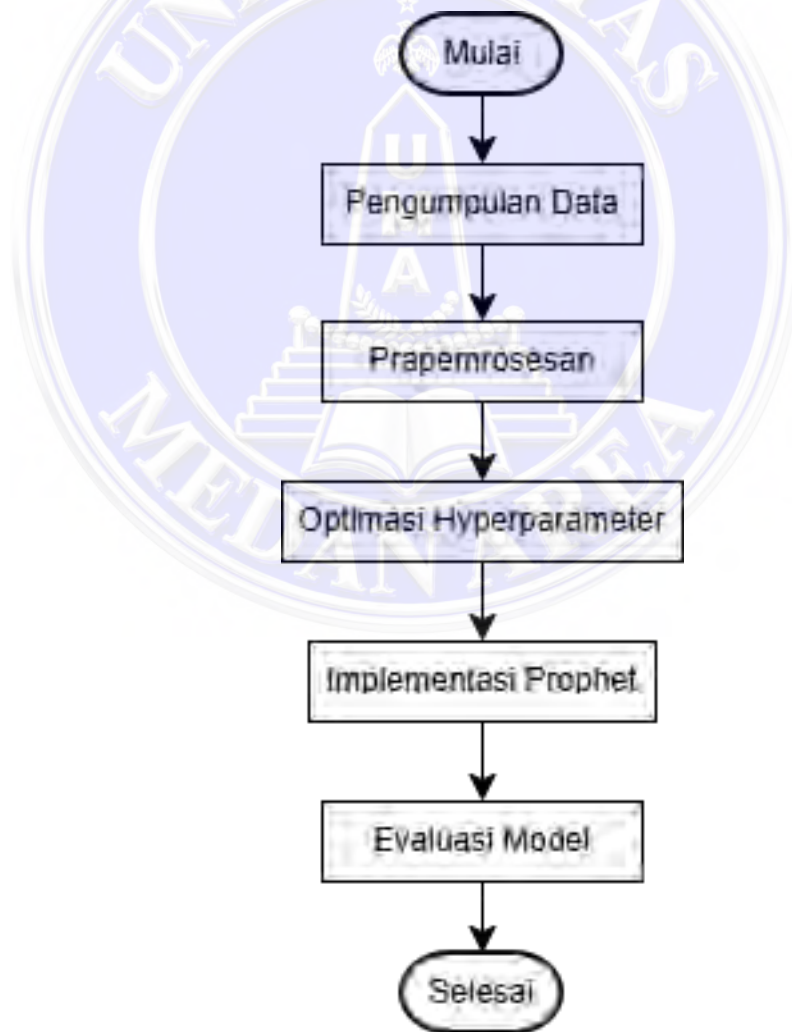
Judul & Peneliti	Metode	Kesimpulan
“Facebook Prophet Model with Bayesian Optimization for USD Index Prediction” (Hamdani dkk., 2023)	Model Prophet dengan optimasi <i>hyperparameter</i> menggunakan <i>Bayesian Optimization</i>	Penelitian ini membuktikan bahwa optimasi Prophet dengan Bayesian Optimization meningkatkan akurasi prediksi indeks USD, dengan penurunan MAPE dari 1,38% menjadi 0,55%, sehingga model lebih adaptif dan stabil terhadap perubahan tren.
“Konfigurasi Model Prophet Untuk Prediksi Harga Saham Sektor Teknologi di Indonesia Yang Akurat” (Santoso & Dewi, 2024)	Prophet	Penelitian ini menggunakan Prophet dengan <i>hyperparameter tuning</i> untuk memprediksi harga saham sektor teknologi. Penyesuaian parameter berhasil menurunkan MAPE hingga 15%, menunjukkan peningkatan akurasi dan efektivitas model.
“Enhancing Prophet Time Series Forecasting on Sparse Data via Hyperparameter Optimisation: A Case Study in Retail” (Atamimi dkk., 2025)	Prophet	Penelitian ini menilai efektivitas Prophet pada data penjualan yang tidak teratur. Melalui penyesuaian manual <i>hyperparameter</i> , MAPE menurun dari 9,50% menjadi 6,80%, menunjukkan bahwa dengan konfigurasi yang tepat, Prophet tetap akurat meski data terbatas.
“Slime Mould Algorithm Training Neural Network in Automatic Voltage Regulator” (Aribowo, 2022)	Neural Network pada sistem Automatic Voltage Regulator (AVR) dioptimasi menggunakan Slime Mould Algorithm (SMA)	Penelitian ini menggunakan SMA untuk melatih jaringan syaraf tiruan pada sistem pengatur tegangan otomatis (AVR). Hasilnya menunjukkan bahwa SMA mempercepat pelatihan, meningkatkan akurasi, dan menghindari jebakan local optima, sehingga menjadi metode optimasi yang efektif dan andal untuk sistem kontrol berbasis neural network.
“Penerapan Algoritma Prophet pada Peramalan Harga Saham” (Simamora dkk., 2024)	Prophet	Prophet mampu memprediksi harga saham UNTR dengan akurasi tinggi, nilai MAPE sebesar 3,86%, sehingga efektif digunakan untuk peramalan data harga saham harian.

BAB III

METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan sistematis yang dirancang untuk memperoleh hasil peramalan yang akurat. Penelitian ini disusun berdasarkan langkah-langkah yang terstruktur, yang dimulai dari pengumpulan data hingga evaluasi hasil akhir. Tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1. *Flowchart Tahapan Penelitian*

Pada Gambar 3.1 diatas adalah *flowchart* tahapan penelitian yang dilakukan Adapun langkah-langkah pada penelitian ini adalah sebagai berikut:

1. Pengumpulan Data: Tahap pertama adalah mengumpulkan data *historis* jumlah pengunjung *website*. Data ini menjadi dasar utama dalam proses analisis dan peramalan.
2. Prapemrosesan: Setelah data terkumpul, langkah selanjutnya adalah melakukan prapemrosesan data. Pada tahap ini, dilakukan pembersihan data dari nilai yang hilang, data *outlier*, maupun anomali yang dapat mempengaruhi kinerja model. Data juga diubah ke dalam format yang sesuai agar mudah diproses oleh algoritma Prophet dan proses optimasi dapat berjalan secara efektif.
3. Optimasi *Hyperparameter*: Pada tahap ini, dilakukan pencarian *hyperparameter* optimal untuk model Prophet menggunakan SMA. SMA merupakan algoritma optimasi metaheuristik yang meniru perilaku jamur lendir dalam mencari makanan, sehingga dapat secara efektif dan efisien menemukan kombinasi *hyperparameter* terbaik yang akan meningkatkan akurasi prediksi.
4. Implementasi Prophet: Setelah *hyperparameter* optimal diperoleh, algoritma Prophet diterapkan untuk membangun model peramalan jumlah pengunjung *website*.
5. Evaluasi: Tahap akhir penelitian ini adalah mengevaluasi kinerja model. Evaluasi dilakukan dengan menggunakan metrik untuk melihat tingkat akurasi hasil prediksi model Prophet yang telah dioptimasi. Hasil evaluasi ini digunakan untuk mengetahui keberhasilan metode yang digunakan.

3.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh dari *platform* Kaggle dengan judul dataset “*Daily Website Visitors*”, yang dapat diakses melalui tautan <https://www.kaggle.com/datasets/bobnau/daily-website-visitors>. Dataset ini berisi informasi mengenai jumlah kunjungan harian *website* materi statistik *forecasting* dalam periode waktu tertentu. Data tersebut digunakan sebagai dasar untuk melatih dan menguji model peramalan jumlah pengunjung *website* menggunakan algoritma Prophet. Secara keseluruhan, dataset ini terdiri dari 2.167 data. Struktur dataset terdiri dari dua kolom utama, yaitu "ds" yang mewakili tanggal pengamatan dalam format *datetime* (hari-bulan-tahun), dan "y" yang menunjukkan jumlah pengunjung *website* pada tanggal tersebut. Tabel 3.1 di bawah ini menampilkan 5 data pertama dan 5 data terakhir dari *Dataset* yang digunakan dalam penelitian ini.

Tabel 3.1. *Dataset* Jumlah Pengunjung Website

No.	Datetime (ds)	First Time Visits(y)
1	9/14/2014	1430
2	9/15/2014	2297
3	9/16/2014	2352
4	9/17/2014	2327
5	9/18/2014	2130
.	.	.
.	.	.
.	.	.
2163	8/15/2020	1,373
2164	8/16/2020	1,686
2165	8/17/2020	2,181
2166	8/18/2020	2,184
2167	8/19/2020	1,297

3.3. Prapemrosesan Data

Prapemrosesan data merupakan tahap awal yang dilakukan sebelum proses pelatihan model, dengan tujuan memastikan bahwa data yang digunakan berada

dalam kondisi bersih, lengkap, dan siap untuk diolah. Tahapan ini sangat penting karena kualitas data secara langsung memengaruhi kinerja dan akurasi model yang dihasilkan.

Langkah-langkah prapemrosesan data meliputi:

1. Pemeriksaan Nilai Hilang (*Missing Value Checking*)

Tahap ini bertujuan untuk mengecek keberadaan nilai kosong dalam dataset.

Nilai yang hilang ditangani dengan beberapa cara, misalnya: menghapus baris atau kolom yang memiliki banyak nilai kosong, atau menggantinya dengan nilai rata-rata (*mean*), median, atau modus sesuai karakteristik data.

2. Pembersihan Data (*Data Cleaning*)

Setelah menangani *missing value*, langkah selanjutnya adalah membersihkan data duplikasi yang dapat dihapus untuk menghindari bias agar tidak memengaruhi proses pelatihan model.

3. Normalisasi Data

Tahap ini dilakukan apabila rentang nilai antar variabel berbeda jauh. Tahap ini menyesuaikan skala data agar fitur memiliki skala seragam. Normalisasi (misalnya *Min-Max Scaling*) mengubah nilai ke rentang 0–1. Normalisasi data dilakukan dengan menggunakan Persamaan (3.1).

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.1)$$

3.4. Optimasi *Hyperparameter*

Tahap optimasi *hyperparameter* merupakan bagian penting dalam pengembangan model peramalan, karena bertujuan untuk menemukan kombinasi

hyperparameter terbaik yang mampu meningkatkan akurasi hasil prediksi. *Hyperparameter* sendiri merupakan parameter yang nilainya tidak dipelajari langsung dari data, melainkan ditetapkan sebelum proses pelatihan model dimulai (Priyanto, 2025). Pemilihan nilai yang tepat sangat berpengaruh terhadap kemampuan model dalam mengenali pola tren maupun musiman pada data historis.

Pada penelitian ini, optimasi difokuskan pada dua *hyperparameter* utama dari model Prophet, yaitu *Changepoint Prior Scale* dan *Seasonality Prior Scale*. *Hyperparameter Changepoint Prior Scale* mengatur tingkat fleksibilitas model dalam menangkap perubahan tren. Nilai yang terlalu kecil membuat model menjadi kaku dan sulit beradaptasi terhadap perubahan mendadak, sedangkan nilai yang terlalu besar dapat menyebabkan model terlalu sensitif terhadap *fluktuasi* data, yang menimbulkan *overfitting*. Sementara itu, *Seasonality Prior Scale* berfungsi mengontrol seberapa besar pengaruh komponen musiman terhadap hasil peramalan. Nilai yang tinggi membuat pola musiman lebih dominan, sedangkan nilai yang terlalu kecil dapat menyebabkan model gagal mengenali pola musiman yang sebenarnya. Penentuan ruang pencarian kombinasi *hyperparameter* dalam penelitian ini mengacu pada penelitian sebelumnya, sehingga rentang nilai awal untuk setiap *hyperparameter* ditetapkan sebagaimana disajikan pada Tabel 3.2 (Hrp dkk., 2025).

Tabel 3.2. Rentang Pencarian

No	<i>Hyperparameter</i>	Rentang Nilai Pencarian
1.	<i>Changepoint Prior Scale</i>	0.001 – 0.5
2.	<i>Seasonality Prior Scale</i>	0.01 – 10

Rentang nilai tersebut digunakan sebagai ruang pencarian (*search space*) dalam proses optimasi menggunakan algoritma SMA untuk menemukan kombinasi

nilai *hyperparameter* yang menghasilkan error terkecil. Untuk memperoleh kombinasi nilai terbaik dari kedua *hyperparameter* tersebut, digunakan pendekatan optimasi dengan algoritma SMA. Melalui mekanisme eksplorasi dan eksploitasi, SMA bekerja secara iteratif untuk mengevaluasi setiap kandidat solusi yang merepresentasikan pasangan nilai *Changepoint Prior Scale* dan *Seasonality Prior Scale*. Evaluasi dilakukan berdasarkan nilai *error* (MAPE), yang mencerminkan perbedaan antara hasil prediksi dan data aktual. Pada setiap iterasi, posisi kandidat solusi diperbarui secara dinamis hingga ditemukan kombinasi nilai yang menghasilkan kesalahan terkecil dan performa model paling optimal.

Hasil dari proses optimasi ini berupa nilai *hyperparameter* terbaik yang selanjutnya digunakan dalam implementasi model Prophet. Dengan adanya optimasi menggunakan SMA, model diharapkan mampu menyesuaikan diri secara lebih adaptif terhadap karakteristik data yang kompleks, sehingga dapat menghasilkan peramalan yang lebih akurat, stabil, dan representatif terhadap kondisi sebenarnya.

3.5. Implementasi Prophet

Pada tahap ini membahas implementasi algoritma Prophet yang digabungkan dengan SMA yang digunakan dalam penelitian untuk meramalkan jumlah pengunjung *website* melalui rancangan *pseudocode*. *Pseudocode* ini disusun untuk menggambarkan alur logika dan tahapan kerja algoritma secara sistematis sebelum diimplementasikan ke dalam kode program. Berikut merupakan *pseudocode* Algoritma Prophet dan SMA:

Pseudocode Algoritma Prophet-SMA:

Input:s

Dataset $D = \{(t_i, y_i) \mid i = 1, 2, \dots, n\}$

Batas parameter:

$cp \in [0.001, 0.5]$

$s \in [0.01, 10.0]$

Ukuran populasi N

Jumlah iterasi maksimum T

Output:

Hyperparameter terbaik(cp_{best}, s_{best})

Hasil peramalan $\hat{y}(t)$

Prosedur Prophet_SMA_Optimization(D):

1. Bagi dataset D menjadi data latih (train) dan data validasi.

2. Inisialisasi populasi SMA:

for $i = 1$ to N:

$cp_i \leftarrow \text{random}(0.001, 0.5)$

$s_i \leftarrow \text{random}(0.01, 10.0)$

$X_i \leftarrow cp_i, s_i$

3. Untuk setiap kandidat X_i

Latih model Prophet menggunakan parameter (cp_i, s_i)

Lakukan peramalan pada data validasi $\rightarrow y_{pred_i}$

Hitung nilai fitness:

$f_i = \text{RMSE}(y_{val}, y_{pred_i})$

4. Tentukan solusi terbaik awal $X_{best}, \arg \min(f_i)$

5. Untuk iterasi $t = 1$ to T:

Urutkan semua X_i berdasarkan nilai fitness (ascending)

Hitung bobot $W_i(|f_i - f_{best}|) + 1$

Untuk setiap kandidat i :

Hasilkan bilangan acak $\in [-1, 1]$

Perbarui posisi:

$X_i = X_i + r \cdot W_i \cdot (X_{best} - X_i)$

Batasi X_i agar tetap berada dalam rentang parameter

Hitung fitness baru $f_i^{new} = \text{RMSE}(\text{Prophet}(X_i))$

If $f_i^{new} < f_i$:

Update $f_i = f_i^{new}$ and $X_i = X_i^{new}$

Perbarui solusi terbaik $X_{best} = \arg \min(f_i)$

6. Kembalikan hyperparameter terbaik $X_{best} = (cp_{best}, s_{best})$

7. Latih model Prophet akhir menggunakan (cp_{best}, s_{best})

Lakukan peramalan jumlah pengunjung di masa depan $\rightarrow \hat{y}(t)$

8. Output $\hat{y}(t)$

End Procedure

3.6. Evaluasi Model

Evaluasi model dilakukan untuk mengetahui seberapa baik model yang dibangun mampu menghasilkan prediksi yang mendekati data sebenarnya. Pada

penelitian ini, proses evaluasi dilakukan terhadap dua model, yaitu Prophet tanpa optimasi dan Prophet yang dioptimasi menggunakan SMA. Tujuannya adalah untuk melihat apakah proses optimasi *hyperparameter* menggunakan SMA benar-benar dapat meningkatkan akurasi hasil peramalan jumlah pengunjung *website*.

Pengujian dilakukan dengan membandingkan hasil prediksi model terhadap data aktual menggunakan metrik kesalahan (*error*) tertentu. Salah satu metrik yang digunakan adalah RMSE. Semakin kecil nilai RMSE, maka hasil prediksi semakin mendekati data sebenarnya. Nilai RMSE yang besar menunjukkan bahwa model belum mampu menggambarkan pola data dengan baik. Selain itu, RMSE juga sensitif terhadap kesalahan besar, sehingga cocok digunakan untuk mengukur seberapa stabil model dalam memprediksi data dengan *fluktuasi* tinggi.

Selain RMSE, penelitian ini juga menggunakan MAPE untuk menilai akurasi dalam bentuk persentase. MAPE menunjukkan seberapa besar rata-rata selisih antara hasil prediksi dengan nilai sebenarnya terhadap nilai aktualnya. Nilai MAPE yang kecil menandakan bahwa model mampu memberikan hasil prediksi yang akurat, sedangkan nilai MAPE yang besar menunjukkan bahwa model masih memiliki tingkat kesalahan yang tinggi.

Selanjutnya dilakukan perbandingan antara model Prophet *default* dan Prophet + SMA berdasarkan hasil nilai RMSE dan MAPE yang diperoleh. Model dengan nilai kesalahan paling kecil dianggap memiliki performa terbaik. Jika hasil menunjukkan bahwa Prophet yang dioptimasi dengan SMA menghasilkan nilai RMSE dan MAPE yang lebih rendah dibandingkan Prophet tanpa optimasi, maka dapat disimpulkan bahwa proses optimasi *hyperparameter* menggunakan SMA berhasil meningkatkan kinerja model.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini bertujuan untuk meningkatkan performa model peramalan jumlah pengunjung *website* berbasis algoritma Prophet melalui optimasi *hyperparameter* menggunakan SMA. Fokus utama penelitian adalah mengoptimalkan *hyperparameter changepoint prior scale* dan *seasonality prior scale* agar model Prophet lebih mampu menyesuaikan pola data deret waktu yang kompleks, khususnya tren, fluktuasi, dan pola musiman pada jumlah pengunjung *website*. Berdasarkan hasil pengujian, tujuan penelitian telah tercapai karena kinerja model meningkat setelah dilakukan optimasi *hyperparameter* menggunakan SMA.

Model Prophet tanpa optimasi menghasilkan nilai RMSE sebesar 399,7137 dan MAPE sebesar 13,5%, sedangkan setelah dilakukan optimasi menggunakan SMA, nilai RMSE berhasil diturunkan menjadi 337,8342 dan MAPE menjadi 11,47%. Penurunan nilai kesalahan ini menunjukkan bahwa model Prophet yang dioptimasi mampu menghasilkan prediksi yang lebih akurat dan lebih mendekati data aktual. Selain itu, hasil visualisasi peramalan menunjukkan bahwa model Prophet + SMA lebih mampu mengikuti fluktuasi data, baik pada periode kenaikan maupun penurunan jumlah pengunjung, serta lebih responsif terhadap perubahan tren dibandingkan model Prophet *default*.

Implikasi dari penelitian ini menunjukkan bahwa penerapan algoritma metaheuristik, khususnya SMA, efektif digunakan sebagai metode optimasi *hyperparameter* pada model Prophet. Secara praktis, kombinasi Prophet dan SMA

dapat menjadi alternatif model peramalan yang lebih akurat dan stabil untuk data deret waktu, terutama dalam peramalan jumlah pengunjung *website*. Selain itu, penelitian ini memberikan kontribusi sebagai referensi dalam pengembangan metode gabungan antara model deret waktu dan algoritma optimasi untuk penelitian selanjutnya.

5.2. Saran

1. Berdasarkan hasil penelitian, masih terdapat beberapa hal yang dapat dikembangkan pada penelitian selanjutnya. Penelitian ini belum membandingkan SMA dengan algoritma metaheuristik lain. Oleh karena itu, penelitian selanjutnya disarankan untuk membandingkan kinerja SMA dengan algoritma lain seperti Genetic Algorithm, PSO, atau Bat Algorithm dalam optimasi *hyperparameter* model Prophet.
2. Penelitian ini juga belum membandingkan model Prophet dengan metode peramalan lainnya. Oleh karena itu, penelitian selanjutnya disarankan untuk membandingkan Prophet yang telah dioptimasi dengan metode lain seperti ARIMA, SARIMA, atau metode berbasis pembelajaran mesin agar keunggulan metode yang digunakan dapat dinilai dengan lebih jelas.
3. penelitian ini belum melakukan uji efisiensi terhadap model yang dikembangkan. Oleh karena itu, penelitian selanjutnya disarankan untuk melakukan pengujian efisiensi komputasi, seperti waktu pemrosesan dan penggunaan sumber daya, guna menilai kelayakan penerapan model pada skala data yang lebih besar.

DAFTAR PUSTAKA

- Amansyah, I., Indra, J., Nurlaelasari, E., & Juwita, A. R. (2024). Prediksi penjualan kendaraan menggunakan regresi linear: Studi kasus pada industri otomotif di indonesia. *Innovative: Journal Of Social Science Research*, 4(4), 1199–1216.
- Anamsyah, H. A., Diyasa, I. G. S. M., & Sihananto, A. N. (2025). Perbandingan Model Xgboost, Lstm, dan Neural Prophet Untuk Prediksi Harga Cabai Rawit Merah Di Jawa Timur. *Prosiding Seminar Nasional Informatika Bela Negara (SANTIKA)*, 5(2), 31–39.
- Anwar, R., & Rasiyanti, L. (2025). Analisis Komparasi Model Peramalan Prophet Dan Arima Dalam Memprediksi Harga Saham Penutupan PT ANTM. *Lattice Journal: Journal of Mathematics Education and Applied*, 5(1), 57–74.
- Aribowo, W. (2022). Slime mould algorithm training neural network in automatic voltage regulator. *Trends in Sciences*, 19(3), 2145.
- Atamimi, F. M. H., Witanti, W., & Abdillah, G. (2025). Enhancing Prophet Time Series Forecasting on Sparse Data via Hyperparameter Optimizattion: A Case Study in Retail. *Sinkron: jurnal dan penelitian teknik informatika*, 9(2), 1000–1007.
- Dai, S., Wang, X., Wang, C., Zhang, H., Ran, J., Zhu, G., & Zhang, J. (2025). Research on the application of Prophet prediction model based on momentum optimization. *Proceedings of the 2025 International Conference on Intelligent Systems, Automation and Control*, 117–122.
- Dinata, R. K., & Hasdyna, N. (2025). *Supervised Learning: Strategi Prediksi dan Klasifikasi Data*. Serasi Media Teknologi.
- Duan, Z., Qian, X., & Song, W. (2025). Multi-Strategy Enhancde Slime Mould

- Algorithm for Optimization Problems. *IEEE Access*.
- Fauzi, J. R. (2020). Algoritma dan flowchart dalam menyelesaikan suatu masalah. *J. Tek. Inform*, 3(2), 12.
- Gharehchopogh, F. S., Ucan, A., Ibrikci, T., Arasteh, B., & Isik, G. (2023). Slime mould algorithm: A comprehensive survey of its variants and applications. *Archives of Computational Methods in Engineering*, 30(4), 2683–2723.
- Hakimah, M., Rahmawati, W. M., & Afandi, A. Y. (2020). Pengukuran Kinerja Metode Peramalan Tipe Exponential Smoothing Dalam Parameter Terbaiknya. *Jurnal Ilmiah NERO*, 5(1).
- Hamdani, A. F., Swanjaya, D., & Helilintar, R. (2023). Facebook Prophet model with Bayesian optimization for USD Index prediction. *JUITA: Jurnal Informatika*, 293–300.
- Hidayat, K., Witanti, W., & Ramadhan, E. (2025). Analisis Tren dan Prediksi Penjualan Restoran Menggunakan Model Time Series Prophet. *METIK JURNAL (AKREDITASI SINTA 3)*, 9(2), 457–467.
- Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions*, 2022, 1–10.
- Hrp, M. R. A., Hasanah, M., Sitepu, F. J., & Lubis, A. H. (2025). Prediksi Harga Emas Antam dengan Menggunakan Algoritma Prophet Melalui Optimasi Hyperparameter Menggunakan Bat Algorithm. *Prosiding Seminar Nasional Teknologi Komputer dan Sains*, 3(1), 394–405.
- Jochumzen, Z., & Skeppstedt, F. (2025). *Machine Learning for Forecasting Order Volumes in Distribution Centres: From Requirements to Implementation*

Under Seasonal and Holiday Demand Variations.

Karunasingha, D. S. K. (2022). Root mean square error or mean absolute error? Use their ratio as well. *Information Sciences*, 585, 609–629.

Kurniawan, T. A. D., Setiawan, A., & Tita, F. (2025). Perbandingan kinerja metode support vector regression dan metode regresi linier berganda dalam memprediksi BMI pada dataset ASTHMA. *Jurnal Sains dan Edukasi Sains*, 8(2), 133–142.

Madhusudhanan, K., Jawed, S., & Schmidt-Thieme, L. (2024). Hyperparameter tuning MLP's for probabilistic time series forecasting. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 264–275.

Naufal, M. (2024). *Peramalan harga emas dengan menggunakan metode autoregressive integrated moving average*. Universitas Islam Negeri Maulana Malik Ibrahim.

Nugraha, D. M. (2025). *Penerapan Analisis Time Series dalam Peramalan Penjualan Berbasis Web (Studi Kasus: Ramen Kaizenka)*. Universitas Kuningan.

Priyanto, P. (2025). *Prediksi hasil panen padi menggunakan Artificial Neural Network berdasarkan faktor cuaca dan kondisi tanah*. Universitas Islam Negeri Maulana Malik Ibrahim.

Qiu, F., Zheng, P., Heidari, A. A., Liang, G., Chen, H., Karim, F. K., Elmannai, H., & Lin, H. (2022). Mutational slime mould algorithm for gene selection. *Biomedicines*, 10(8), 2052.

Rozy, A. F., Solimun, S., & Wardhani, N. W. S. (2023). Peningkatan Akurasi Metode Weighted Fuzzy Time Series Forecasting Menggunakan Algoritma

- Evolusi Differensial dan Fuzzy C-Means. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 10(5), 1047–1054.
- Rusman, J., Haryati, B. Z., Michael, A., & others. (2023). Optimisasi hiperparameter tuning pada metode Support Vector Machine untuk klasifikasi tingkat kematangan buah kopi. *J-Icon: Jurnal Komputer dan Informatika*, 11(2), 195–202.
- Santoso, R. S., & Dewi, F. K. S. (2024). Konfigurasi model Prophet untuk prediksi harga saham sektor teknologi di Indonesia yang akurat. *Jurnal Buana Informatika*, 15(01), 50–58.
- Simamora, J. P., Martha, S., & Andani, W. (2024). Penerapan algoritma prophet pada peramalan harga saham. *Equator: Journal of Mathematical and Statistical Sciences*, 3(2), 94–101.
- Smrti, N. N. E., Sukenada, A. I. P. G., Kadek, D. T. R. N., Adnan, A., & Ode, J. P. P. (2023). Flowgorithm Sebagai Penunjang Pembelajaran Algoritma dan Pemrograman. *Jurnal Bangkit Indonesia*, 12(1), 56–64.
- Syaliman, K. U., Maysofa, L., & Sapriadi, S. (2023). Implementasi forecasting pada penjualan inaura hair care dengan metode single exponential smoothing. *Jurnal Testing dan Implementasi Sistem Informasi*, 1(2), 82–91.
- Tang, A.-D., Tang, S.-Q., Han, T., Zhou, H., & Xie, L. (2021). A modified slime mould algorithm for global optimization. *Computational intelligence and neuroscience*, 2021(1), 2298215.
- Taofiqurrohman, H., Wufron, W., & Roji, F. F. (2025). Prediksi Harga Saham Telkom Menggunakan Prophet: Analisis Pengaruh Sentimen Publik Terhadap Kehadiran Starlink: Telkom Stock Price Prediction Using Prophet: Analysis

of the Effect of Public Sentiment on the Presence of Starlink. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 5(2), 484–495.

Wei, Y., Othman, Z., Daud, K. M., Luo, Q., & Zhou, Y. (2024). Advances in slime mould algorithm: a comprehensive survey. *Biomimetics*, 9(1), 31.

Wiejaya, A., & Fenriana, I. (2024). Prediksi Harga Saham Top 10 NASDAQ dengan Time Series Prophet. *bit-Tech*, 7(2), 252–262.

Yao, E., Zhang, L., Li, X., & Yun, X. (2023). Traffic forecasting of back servers based on ARIMA-LSTM-CF hybrid model. *International Journal of Computational Intelligence Systems*, 16(1), 65.

Zikri, M. A. (2024). Pemanfaatan Pemodelan Neural Prophet dalam Memprediksi Parameter Pm10 (Studi Kasus Dki Jakarta). *Megasains*, 15(1), 44–50.

LAMPIRAN

1. Kode Program

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from prophet import Prophet
from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error

def prophet_fitness(params):
    cp, sp = params

    cp = np.clip(cp, 0.001, 0.5)
    sp = np.clip(sp, 0.01, 10.0)

    model = Prophet(
        changepoint_prior_scale=cp,
        seasonality_prior_scale=sp,
        yearly_seasonality=True,
        weekly_seasonality=True
    )

    model.fit(train_df)
    future = model.make_future_dataframe(periods=len(test_df))
    forecast = model.predict(future)

    y_pred = forecast['yhat'].iloc[-len(test_df):].values
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    mape = mean_absolute_percentage_error(y_true, y_pred)

    return 0.5 * rmse + 0.5 * mape

def slime_mould_algorithm(fitness_func, lb, ub, pop_size=8, max_iter=20):
    dim = len(lb)
    pop = np.random.uniform(lb, ub, (pop_size, dim))
    fitness = np.array([fitness_func(ind) for ind in pop])

    best_idx = np.argmin(fitness)
    best_sol = pop[best_idx]
    best_fit = fitness[best_idx]

    for t in range(max_iter):
        idx = np.argsort(fitness)
        pop, fitness = pop[idx], fitness[idx]

        epsilon = 1e-6
        a = np.arctanh(1 - (t / max_iter) - epsilon)

        for i in range(pop_size):
            if np.random.rand() < 0.5:
                diff = pop[np.random.randint(pop_size)] - pop[np.random.randint(pop_size)]
                pop[i] = best_sol + a * diff
            else:
                pop[i] = lb + np.random.rand(dim) * (ub - lb)
```

```

pop[i] = np.clip(pop[i], lb, ub)
pop[i] = np.nan_to_num(pop[i], nan=0.01, posinf=0.5, neginf=0.001)

fitness[i] = fitness_func(pop[i])

if fitness.min() < best_fit:
    best_fit = fitness.min()
    best_sol = pop[np.argmin(fitness)]

print(f"Iterasi {t+1}/{max_iter} | Best Fitness: {best_fit:.4f}")

return best_sol
lb = np.array([0.001, 0.01])
ub = np.array([0.5, 10.0])

best_params = slime_mould_algorithm(prophet_fitness, lb, ub)

print("Parameter Optimal:")
print("Changepoint Prior Scale :", best_params[0])
print("Seasonality Prior Scale :", best_params[1])
model_sma = Prophet(
    changepoint_prior_scale=best_params[0],
    seasonality_prior_scale=best_params[1],
    yearly_seasonality=True,
    weekly_seasonality=True
)

model_sma.fit(train_df)

future = model_sma.make_future_dataframe(periods=len(test_df))
forecast_sma = model_sma.predict(future)
y_pred_sma = forecast_sma['yhat'].iloc[-len(test_df):].values

rmse_sma = np.sqrt(mean_squared_error(y_true, y_pred_sma))
mape_sma = mean_absolute_percentage_error(y_true, y_pred_sma)

print("=== HASIL EVALUASI PROPHET + SMA ===")
print(f"RMSE : {rmse_sma:.4f}")
print(f"MAPE : {mape_sma:.4f}")

```