

Implementasi Metode *Prophet* pada Prediksi Tinggi Air Sungai

1st Achmad Raihan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

achmadraihan@student.telkomuniversity.ac.id

2nd Asep Suhendi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

suhendi@telkomuniversity.ac.id

3rd Hertiana Bethaningtyas
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

hertiana@telkomuniversity.ac.id

Abstrak—Banjir merupakan salah satu peristiwa yang menghasilkan dampak buruk besar hingga mengakibatkan berhentinya berbagai sektor aktivitas kehidupan manusia. Dalam penanggulangannya, terdapat berbagai cara telah dilakukan yang satunya membuat sistem prediksi tinggi air sungai menggunakan *machine learning*. Namun, berdasarkan penelitian tersebut, metode evaluasi yang digunakan yaitu tidak dapat melakukan prediksi ke masa depan. Selain itu tidak memiliki sebuah *display* yang dapat menampilkan hasil prediksi tinggi air sungai. Tugas Akhir ini mengusulkan untuk dapat mengimplementasikan metode *prophet* pada prediksi tinggi air sungai. Kemudian, sistem tersebut dapat menampilkan hasil prediksi tinggi air sungai dalam sebuah *dashboard*. Selanjutnya, data set yang digunakan untuk prediksi banjir hanya hasil akhir pengukuran tinggi air sungai setiap 1 jam dengan mengabaikan faktor yang mempengaruhinya. Evaluasi setelah dilakukan *hyperparameter tuning* dengan 3 data berbeda menggunakan *Theil's U* menghasilkan rata-rata *error* terkecil yaitu 0.187044. *Dashboard* dapat memprediksi dan menampilkan data secara langsung setelah *upload* dilakukan. Kemudian, *dashboard* memiliki contoh data yang dapat digunakan untuk melihat data yang dapat ditampilkan setelah dilakukan prediksi.

Kata kunci— *prophet*, *forecast*, *dashboard*

I. PENDAHULUAN

Banjir merupakan salah satu peristiwa yang menghasilkan dampak buruk besar hingga mengakibatkan berhentinya berbagai sektor aktivitas kehidupan manusia [1]. Hal itu disebabkan karena luapan dimana sungai tidak dapat menampung air yang dihasilkan [2]. Pada wilayah perkotaan, pembangunan infrastruktur berskala besar tanpa perencanaan yang bagus memicu terjadinya banjir ketika dangkalnya kedalaman air sungai.

Terdapat berbagai penanggulangan yang telah dilakukan, salah satunya dengan membuat sistem prediksi tinggi air sungai menggunakan *machine learning*. Dalam pembuatannya, berbagai model *machine learning* telah digunakan antara lain *Single Exponential Smoothing*, *Brown's Exponential Smoothing* [3], *Autoregressive Integrated Moving Average*, *Autoregressive*, *Autoregressive Moving Average* [4]. Namun, berdasarkan penelitian tersebut, metode evaluasi yang digunakan yaitu tidak dapat melakukan prediksi ke masa depan. Selain itu tidak memiliki sebuah *display* yang dapat menampilkan hasil prediksi tinggi air sungai.

Tugas Akhir ini mengusulkan untuk dapat mengimplementasikan metode *prophet* pada prediksi tinggi

air sungai. *Prophet* digunakan karena dapat melakukan prediksi dengan cara yang lebih mudah dilakukan, masuk akal, dan akurat. Selain itu, *prophet* fleksibel terhadap sistem prediksi yang dibuat sehingga dapat digunakan oleh berbagai kalangan ahli dengan cara intuitif [5]. Dengan adanya Tugas Akhir ini, diharapkan dapat mengimplementasikan metode *prophet* pada prediksi tinggi air sungai.

II. KAJIAN TEORI

A. Machine Learning

Machine learning merupakan salah satu bidang *artificial intellegent* yang dapat digunakan untuk memetakan suatu pola data dengan cara melakukan pelatihan, pengujian, dan evaluasi. *Machine learning* dapat memproses suatu data dengan performa tinggi yang menyediakan kemudahan dalam penerapannya [1]. Terdapat 2 faktor mempengaruhi keberhasilan sistem yang dibuat, yaitu efektifitas model statistik dalam memperoleh data kompleks dan skalabilitas sistem dalam mempelajari keterkaitan model dari *dataset* [6]. Dalam *machine learning*, terdapat 3 kategori untuk melakukan prediksi berdasarkan jenis data, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

B. Machine Learning dalam Melakukan Time Series Forecasting

Machine learning telah banyak digunakan di berbagai bidang sektor kehidupan. Penggunaan tersebut bergantung pada kebutuhan setiap desain sistem dan *data set*. Oleh karena itu, dalam pengerjaan tugas akhir, dibuat sistem prediksi tinggi air sungai menggunakan model *prophet* untuk data bersifat regresi dengan kondisi *time series*.

Time series merupakan *data set* dengan variabel sama yang telah diamati selama periode waktu tertentu secara berurutan. Data tersebut sering dibuat dalam bentuk grafik garis dengan waktu sebagai *x-axis* dan *values* merupakan *y-axis*. Terdapat 4 komponen dasar dalam *time series*, yaitu *trend*, *seasonality*, *cyclic*, dan *random variations* atau *irregular* [7].

Prophet merupakan sebuah algoritma *machine learning* bersifat *supervised learning* kategori regresi yang dibuat oleh para peneliti dari Perusahaan Meta untuk dapat menyelesaikan berbagai kasus data berkaitan dengan *time series* [5].

$$y(t) = g(t) + s(t) + h(t) + \epsilon(t) \quad 2.1$$

$g(t)$ merupakan fungsi *trend* yang mana model secara non-periodik dapat mengalami perubahan nilai terhadap *time series*. $s(t)$ merupakan fungsi *trend* yang mana model secara periodik dapat mengalami perubahan nilai terhadap *time series*, misalnya dalam seminggu atau setahun. $h(t)$ merupakan fungsi yang merepresentasikan dampak kejadian di waktu khusus seperti hari libur. $\epsilon(t)$ merupakan perubahan khusus yang terjadi ketika tidak terakomodasi oleh model.

1. Parameter untuk Prediksi Tinggi Air Sungai

Terdapat banyak parameter yang dapat digunakan dalam melakukan *hyperparameter tuning* untuk prediksi tinggi air sungai. Meskipun begitu, hal tersebut bukan parameter utama dalam melakukan *hyperparameter tuning*. Dalam hal ini, terdapat beberapa parameter utama yang disediakan oleh *prophet* ketika melakukan *hyperparameter tuning*. Hal itu disebabkan karena pengaruh yang dihasilkan dari parameter tersebut memiliki dampak besar terhadap hasil akhir pada model tersebut. Berikut ini merupakan parameter yang direkomendasikan untuk dilakukan *hyperparameter tuning* dalam suatu model oleh *prophet* [8].

a. *Changepoint Prior Scale*

Changepoint prior scale merupakan parameter yang menentukan fleksibilitas dari suatu *trend*, khususnya seberapa banyaknya perubahan pada *trend changepoints*. Penggunaan nilai-nilai parameter yang semakin tinggi dapat menghasilkan fleksibilitas yang tinggi. Begitupun sebaliknya, Penggunaan nilai-nilai parameter yang semakin rendah dapat menghasilkan *noise* pada model. Dalam pembuatan prediksi tinggi air sungai, proses *hyperparameter tuning* pada *changepoint prior scale* dilakukan pada nilai 0,1 0,2 0,3 0,4 0,5 [8].

b. *Seasonality Prior Scale*

Seasonality prior scale merupakan parameter yang mengontrol fleksibilitas dari *seasonal*. Penggunaan nilai-nilai parameter yang semakin tinggi dapat menghasilkan fleksibilitas yang bagus. Begitupun sebaliknya, Penggunaan nilai-nilai parameter yang semakin rendah dapat menghasilkan *magnitude* kecil. Dalam pembuatan prediksi tinggi air sungai, proses *hyperparameter tuning* pada *seasonality prior scale* dilakukan pada nilai 6 7 8 9 10 [8].

c. *Seasonality Mode*

Seasonality mode merupakan parameter yang memberikan pilihan dalam menentukan pola *trend*, yaitu antara *additive* atau *multiplicative*. *Additive* merupakan pola *trend* yang perubahan nilainya bersifat cenderung statis seiring berjalannya waktu. Sedangkan *multiplicative* merupakan pola *trend* yang perubahan nilainya bersifat cenderung dinamis seiring berjalannya waktu. Pilihan *default* pada *seasonality mode* yaitu *additive*. Ketika *hyperparameter tuning* dilakukan, akan dipilih salah satu diantara keduanya yang terbaik untuk digunakan dalam model [8].

C. *Short Term Forecasting*

Prediksi ke masa depan dilakukan dengan menggunakan *short term forecasting*. *Short term forecasting* merupakan cara untuk melakukan prediksi ke masa depan dengan jangka pendek berdasarkan pengaturan durasi waktu yang ditentukan. Berdasarkan hal tersebut, dengan *data set*

berdurasi kurang dari 2 tahun dan intervalnya 1 jam, maka prediksi dapat dilakukan hingga 7 hari ke depan [9].

D. *Theil's U* untuk Evaluasi Pembuatan Model *Prophet*

Theil's U merupakan cara yang digunakan untuk mengukur akurasi pada pola permasalahan regresi. *Theil's U* memiliki rentang nilai 0 hingga 1. Semakin mendekati 0 nilai *error* maka model yang dibuat bagus, begitupun sebaliknya. A , P di persamaan 5 merupakan nilai aktual dan prediksi dalam melakukan perhitungan. Kemudian, n merupakan jumlah data yang digunakan dalam melakukan perhitungan [10].

E. GitHub sebagai *File Repository* dalam *Deploy Model Prophet*

GitHub merupakan *platform* yang digunakan para *developer* untuk dapat melakukan manajemen proyek dari berbagai *source code* dan sebagai jejaring sosial dalam melakukan pekerjaan bersifat jarak jauh. GitHub menawarkan 3 hal kepada *developer* untuk dapat menunjang pekerjaannya, yaitu produktivitas, kolaborasi, dan keamanan. Dengan adanya GitHub, dapat menunjang produktivitas seseorang dalam mengembangkan perangkat lunak dengan kualitas tinggi dan meningkatkan kecepatan dalam berinovasi. Kemudian, GitHub dapat membuat seseorang untuk berkolaborasi dengan orang lain dalam melakukan pekerjaannya meskipun dalam kondisi jarak jauh. Terakhir, dengan adanya fitur berupa dapat menyimpan *source code* ke sebuah *repository*, maka GitHub dapat menjamin keamanan setiap *file* yang tersimpan, baik itu dapat diatur untuk dapat diakses semua orang atau hanya dalam kondisi tertentu. Dalam prediksi tinggi air sungai, GitHub digunakan untuk dapat menyimpan *source code* dan contoh data untuk diintegrasikan ke *Streamlit*. Hal itu dilakukan agar *dashboard* dapat menampilkan data berbasis *website* [13] [14].

F. *Streamlit* sebagai *Dashboard* Prediksi Tinggi Air Sungai

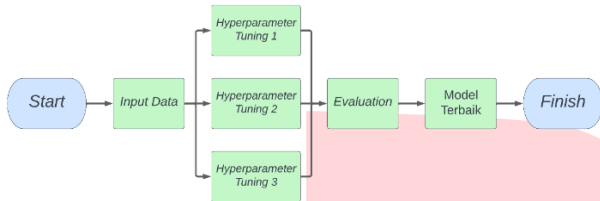
Dashboard merupakan wadah untuk dapat mengintegrasikan kinerja prediksi tinggi air sungai. Ketika model untuk prediksi tinggi air sungai telah dibuat, dilakukan *deploy* ke dalam bentuk *website* yang dapat diakses oleh banyak orang. Kemudian, model tersebut di *deploy* menggunakan *Streamlit*. *Streamlit* merupakan *platform* yang banyak digunakan dalam kasus *machine learning* dan *data science* ketika melakukan *deploy* untuk dapat menampilkan hasil prediksi dan analisis berbasis *website* [11]. Terdapat beberapa keuntungan yang diperoleh dengan menggunakan *platform* *Streamlit* dalam pembuatan *dashboard*. Pertama, dengan menggunakan bahasa pemrograman yaitu *Python*, proses integrasi untuk dapat ditampilkan dalam bentuk *website* akan semakin mudah dilakukan. Kedua, kemudahan dalam pembuatan fitur di *user interface* pada *website* [12].

III. METODE

A. Pembuatan Model *Prophet*

Pembuatan model *prophet* dilakukan dengan cara *hyperparameter tuning*. *Hyperparameter tuning* dilakukan dengan cara melakukan pengaturan nilai berdasarkan parameter yang dapat digunakan untuk membuat sistem prediksi tinggi air sungai. *Hyperparameter tuning* dilakukan sebanyak 3 kali dengan *data set* yang berbeda-beda. *Data set*

pertama berisikan tinggi air sungai sejak tanggal 1 Januari 2020 hingga 30 Juni 2020. *Data set* kedua berisikan tinggi air sungai sejak tanggal 1 Juli 2020 hingga 31 Desember 2020. *Data set* ketiga berisikan tinggi air sungai sejak tanggal 1 April 2020 hingga 30 September 2020. Dalam pembuatan prediksi tinggi air sungai, *hyperparameter tuning* bertujuan untuk menentukan model terbaik dengan performa yang tinggi. Kemudian, dilakukan evaluasi dengan menggunakan *theil's u* untuk mengetahui model terbaik dengan cara melihat *error* terkecil. Setelah itu dapat ditentukan model terbaik untuk diimplementasikan ke *dashboard* untuk prediksi tinggi air sungai.



Gambar 3.1 Pembuatan Model Prophet

B. PEMBUATAN DASHBOARD PREDIKSI TINGGI AIR SUNGAI

1. Deploy Model Terbaik Prophet ke Streamlit

Deploy model terbaik *prophet* ke Streamlit bertujuan untuk dapat menampilkan kinerja dalam bentuk *dashboard*. Sebelum dapat melakukan *deploy*, *developer* harus memiliki akun GitHub. Kemudian memiliki akun Streamlit yang sudah terintegrasi dengan GitHub. Hal tersebut penting dilakukan karena merupakan syarat awal sebelum melakukan *deploy* model terbaik *prophet* ke Streamlit. Ketika *deploy* telah dilakukan, Streamlit akan membaca *code* yang dibuat berdasarkan GitHub *repository*.



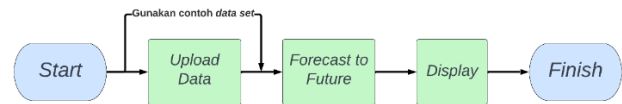
Gambar 3.2 Alur Deploy Model Terbaik Ke Streamlit

Integrasi model terbaik ke *streamlit* secara *local* merupakan tahap dimana *backend* dan *frontend* disatukan secara utuh tetapi kinerjanya hanya bisa dilihat oleh *developer*. Ini merupakan tahap awal untuk mengetahui kinerja *dashboard* sebelum dilakukan *deploy*. Dengan begitu, ketika telah dilakukan *deploy* diharapkan *dashboard* tidak memiliki kendala ketika melakukan pekerjaannya.

Berikutnya, letakkan *file* hasil integrasi secara *local* tersebut ke dalam sebuah GitHub *repository*. Terdapat 3 *file* yang harus diletakkan ke sebuah GitHub *repository*. Pertama, *app.py* merupakan *file* berisikan *code* yang sudah mengintegrasikan model terbaik dengan Streamlit. Kedua, *requirements.txt* merupakan *file* berisikan *library package* yang digunakan ketika membuat prediksi tinggi air sungai. Ketiga, *kalu.csv* merupakan *data set* tinggi air sungai. *Data set* tersebut dilampirkan untuk dapat menunjukkan kinerja *dashboard* tanpa melakukan *upload data*.

Integrasi GitHub *repository* ke Streamlit merupakan tahap terakhir dalam melakukan *deploy*. Pada tahap ini, *developer* dapat mengunjungi <https://share.streamlit.io/> dimana merupakan situs untuk *deploy* model terbaik *prophet* ke Streamlit. Setelah integrasi dilakukan, koneksi internet harus dalam kondisi stabil untuk keamanan dan kelancaran *deploy* model terbaik *prophet* ke Streamlit.

2. Proses Kerja Dashboard



Gambar 3.3 Alur Kerja Dashboard

Upload data dapat disebut sebagai tahap *input data set* ke *dashboard*. Sebelum dilakukan *upload data set* tinggi air sungai, penamaan kolom pada *file* harus dikondisikan untuk variabel *independent* dan *dependent*. Variabel *independent* diberi nama *ds* dan *dependent* yaitu *y*. Penamaan variabel tersebut disebabkan karena merupakan bagian dari cara penggunaan model *prophet* pada *data set* ketika melakukan pekerjaannya. Tetapi, jika tidak memiliki *data set* berupa tinggi air sungai, pengguna dapat menekan tombol 'Gunakan contoh data set' untuk melihat kinerja *dashboard* tanpa melakukan *upload data*.

Forecast to future merupakan tahap dimana pengguna dapat mengatur rentang waktu untuk melakukan prediksi ke masa depan. Prediksi ke masa depan maksimal dilakukan selama 7 hari ke depan karena dilakukan menggunakan cara *short term forecasting*. Meskipun durasi maksimal 7 hari, interval waktu yang digunakan yaitu tiap 1 jam. Dengan begitu, *dashboard* dapat melakukan prediksi ke depan sebanyak 168 data.

Display merupakan tahap dimana *dashboard* menampilkan data tinggi air sungai. *Dashboard* dapat menampilkan 2 hal, yaitu grafik data asli serta hasil prediksi tinggi air sungai. Hasil tersebut termasuk sudah dapat menampilkan prediksi ke masa depan tinggi air sungai.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pembuatan Model Prophet

Pembuatan model dilakukan dengan cara melakukan *hyperparameter tuning* sebanyak 3 kali dengan *data set* berbeda-beda. Berdasarkan hasil tersebut, dipilih 5 model terbaik dari setiap *hyperparameter tuning* yang telah dilakukan.

Tabel 4.1 Hasil hyperparameter tuning pada data pertama

No.	Changepoint Prior Scale	Seasonality Prior Scale	Seasonality Mode	Theil's U
1	0.5	7.0	multiplicative	0.193935
2	0.5	9.0	multiplicative	0.194098
3	0.5	10.0	multiplicative	0.194119
4	0.5	8.0	multiplicative	0.194159
5	0.5	6.0	multiplicative	0.194218

Tabel 4.2 Hasil hyperparameter tuning pada data kedua

No.	Changepoint Prior Scale	Seasonality Prior Scale	Seasonality Mode	Theil's U
1	0.5	8.0	multiplicative	0.156753
2	0.5	9.0	multiplicative	0.156884
3	0.5	7.0	multiplicative	0.156926
4	0.5	10.0	multiplicative	0.156935
5	0.5	6.0	multiplicative	0.156977

Tabel 4.3 Hasil hyperparameter tuning pada data ketiga

No.	Changepoint Prior Scale	Seasonality Prior Scale	Seasonality Mode	Theil's U
1	0.5	10.0	multiplicative	0.210212

2	0.5	6.0	<i>multiplicative</i>	0.210242
3	0.5	8.0	<i>multiplicative</i>	0.210251
4	0.5	7.0	<i>multiplicative</i>	0.210271
5	0.5	9.0	<i>multiplicative</i>	0.210271

Berdasarkan hasil tersebut, akan dipilih 3 terbaik dari 5 model setiap *hyperparameter tuning* yang telah dilakukan. Setiap pemilihan model terbaik dipilih berdasarkan *error* terkecil. Hal itu dilakukan karena setiap *hyperparameter tuning* menghasilkan nilai parameter terbaik yang berbeda-beda.

Tabel 4.4
Model terbaik dari setiap *hyperparameter tuning*

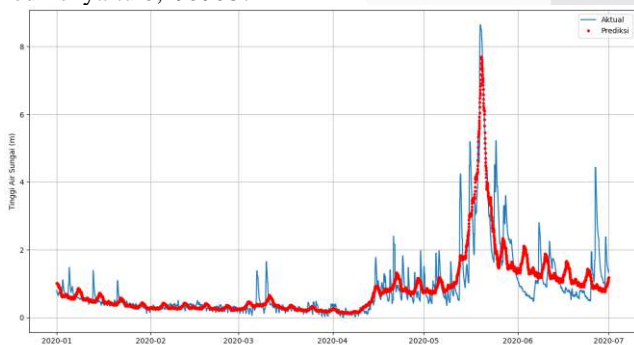
<i>Changeoint Prior Scale</i>	<i>Seasonality Prior Scale</i>	<i>Seasonality Mode</i>	<i>Theil's U Data 1</i>	<i>Theil's U Data 2</i>	<i>Theil's U Data 3</i>
0.5	10.0	<i>multiplicative</i>	0.194119	0.156935	0.210212
0.5	8.0	<i>multiplicative</i>	0.194159	0.156753	0.210251
0.5	7.0	<i>multiplicative</i>	0.193935	0.156926	0.210271

Berdasarkan tabel 4.4, model terbaik ditentukan berdasarkan rata-rata *error* terkecil. Model terbaik yaitu dengan *error* terkecil dapat dilihat pada baris pertama tabel 4.5. Dengan begitu, model dapat diimplementasikan ke dalam sistem prediksi tinggi air sungai.

Tabel 4.5
Model terbaik berdasarkan rata-rata *error* terkecil

<i>Changeoint Prior Scale</i>	<i>Seasonality Prior Scale</i>	<i>Seasonality Mode</i>	Rata-Rata <i>Theil's U</i>
0.5	7.0	<i>multiplicative</i>	0.187044
0.5	8.0	<i>multiplicative</i>	0.18705433
0.5	10.0	<i>multiplicative</i>	0.18708867

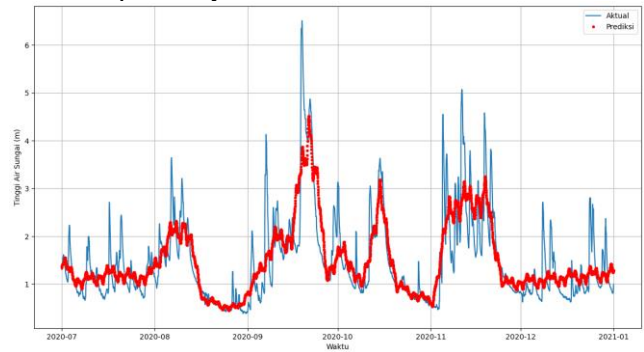
Pada gambar 4.1, nilai aktual diberi warna biru. Sedangkan nilai prediksi diberi warna merah. Kemudian, model terbaik berusaha membaca nilai aktual untuk dapat melakukan prediksi hingga sedekat mungkin. Selain itu, terdapat kemiripan pada nilai aktual dalam beberapa bulan awal. Tetapi, pada beberapa bulan terakhir terdapat fluktuatif berskala kecil dalam jumlah banyak dan terdapat 1 gelombang dengan amplitudo mencolok pada nilai aktual. Meskipun begitu, model terbaik tetap dapat membaca *trend* pada nilai aktual dari *data set* untuk dilakukan prediksi. Dengan begitu, *error* yang dihasilkan ketika melakukan prediksi yaitu 0,193935.



Gambar 4.1
Grafik model terbaik menggunakan data pertama

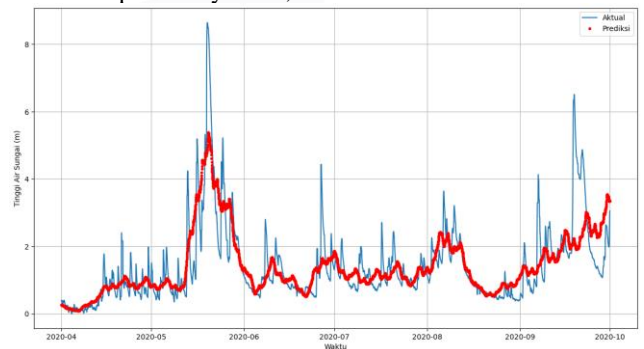
Pada gambar 4.2, nilai aktual diberi warna biru. Sedangkan nilai prediksi diberi warna merah. Kemudian, model terbaik berusaha membaca nilai aktual untuk dapat melakukan prediksi hingga sedekat mungkin. Tetapi, terdapat data fluktuatif berskala besar dalam jumlah cukup banyak pada nilai aktual. Model terbaik cenderung membaca

fluktuatif tersebut sebagai *outlier* dalam sebuah *trend*. Dengan demikian, ketika dilihat dari *trend*, model terbaik dapat membacanya hingga *error* yang dihasilkan dalam melakukan prediksi yaitu 0,156926.



Gambar 4.2
Grafik model terbaik menggunakan data ketiga

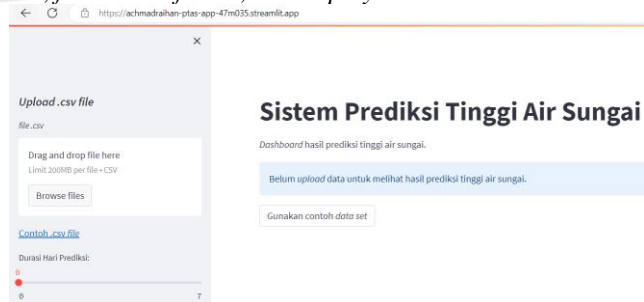
Pada gambar 4.3, nilai aktual diberi warna biru. Sedangkan nilai prediksi diberi warna merah. Kemudian, model terbaik berusaha membaca nilai aktual untuk dapat melakukan prediksi hingga sedekat mungkin. Tetapi, terdapat data fluktuatif yang besar dalam jumlah banyak pada nilai aktual. Kemudian, jika dilihat dari amplitudo dan lembah pada data aktual, menghasilkan fluktuatif dalam jumlah lebih banyak dibanding data kedua dengan skala yang kecil. Meskipun begitu, ketika dilihat dari *trend*, model terbaik dapat membacanya hingga *error* yang dihasilkan dalam melakukan prediksi yaitu 0,210271.



Gambar 4.3
Grafik model terbaik menggunakan data ketiga

B. Menampilkan Data dengan Menggunakan *Dashboard*

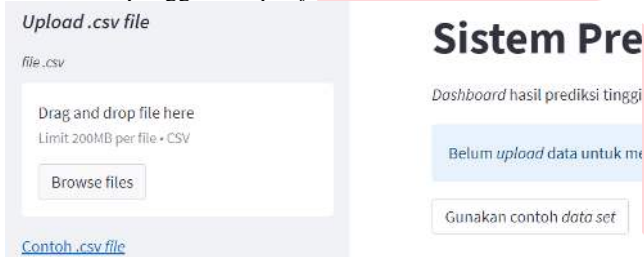
Model terbaik yang telah dipilih akan diimplementasikan ke *dashboard*. *Dashboard* dibuat untuk dapat menampilkan hasil prediksi tinggi air sungai. Terdapat 3 komponen utama dalam *dashboard* prediksi tinggi air sungai, yaitu *upload data*, *forecast to future*, dan *display*.



Gambar 4.4
Tampilan awal *website*

Upload data dapat disebut sebagai tahap *input data set* ke *dashboard*. Dalam *dashboard* prediksi tinggi air sungai,

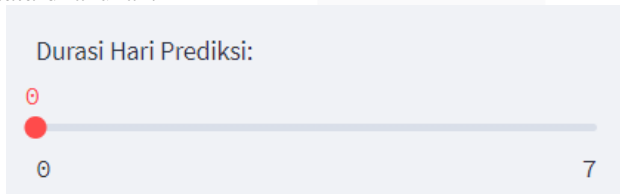
posisinya terdapat di atas pojok kiri. *File* untuk *upload data* harus dalam format *.csv*. Kemudian, sebelum dilakukannya *upload data set* tinggi air sungai, penamaan kolom pada *file* harus dikondisikan untuk variabel *independent* dan *dependent*. Variabel *independent* diberi nama *ds* dan *dependent* yaitu *y*. Penamaan variabel tersebut disebabkan karena merupakan bagian dari cara penggunaan model *prophet* pada *data set* ketika melakukan pekerjaannya. Tetapi, jika tidak memiliki *data set* berupa tinggi air sungai, pengguna dapat menekan tombol 'Gunakan contoh *data set*' untuk melihat kinerja *dashboard* tanpa melakukan *upload data*. Kemudian, terdapat link 'Contoh *.csv file*' bertujuan untuk melihat contoh *file* yang dapat digunakan dalam *dashboard* prediksi tinggi air sungai. Pada *dashboard*, batas maksimum ukuran *file* yang dapat di *upload* yaitu 200 MB berdasarkan penggunaan platform Streamlit.



Gambar 4.5

Fitur *upload data* pada *dashboard*

Forecast to future merupakan tahap dimana pengguna dapat mengatur rentang waktu untuk melakukan prediksi ke masa depan. Posisi pengaturan prediksi ke masa terdapat dibawah *upload data* pada *dashboard*. Pengguna dapat mengatur rentang waktu prediksi ke masa depan dengan cara menggeser ke kanan atau ke kiri setelah atau sebelum *upload data* dilakukan.



Gambar 4.6

Fitur *forecast to future* pada *dashboard*

Display merupakan tahap dimana *dashboard* menampilkan data tinggi air sungai. *Display* akan bekerja setelah *upload data* dilakukan atau tombol 'Gunakan contoh *data set*' ditekan. *Dashboard* dapat menampilkan 2 hal, yaitu grafik data asli serta hasil prediksi tinggi air sungai. Hasil tersebut termasuk sudah dapat menampilkan prediksi ke masa depan tinggi air sungai.

Grafik Hasil Prediksi Tinggi Air Sungai 5 Hari



Gambar 4.7
Display hasil prediksi tinggi air sungai

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, disimpulkan bahwa model *prophet* yang digunakan dapat diimplementasikan ke dalam sistem prediksi tinggi air sungai setelah dilakukan *hyperparameter tuning* menggunakan 3 data berbeda. Evaluasi menggunakan *Theil's U* menghasilkan rata-rata *error* terkecil yaitu 0.187044. Kemudian, *dashboard* dapat menampilkan hasil prediksi tinggi air sungai. *Dashboard* dapat memprediksi dan menampilkan data secara langsung setelah *upload* dilakukan. Kemudian, *dashboard* memiliki contoh data yang dapat digunakan untuk melihat data yang dapat ditampilkan setelah dilakukan prediksi.

REFERENSI

- [1]. Mosavi, A, Ozturk, P, & Chau, K (2018). Flood prediction using machine learning models: Literature review. *Water*, mdp.com, <https://www.mdpi.com/357544>
- [2]. Kartika, Ni Komang Ega (2020). *PREDIKSI BANJIR MENGGUNAKAN RADIAL BASIS FUNCTION (RBF) BERBASIS INTERNET OF THINGS (IOT)*., Universitas Telkom, S1 Teknik Elektro, <https://openlibrary.telkomuniversity.ac.id/pustaka/164414/prediksi-banjir-menggunakan-radial-basis-function-rbf-berbasis-internet-of-things-iot.html>
- [3]. Vlandari, RT, Wijayanto, H, & Laksito, W (2019). Perbandingan Prediksi Tinggi Muka Air Bendungan Wonogiri dengan Single Exponential Smoothing dan Brown's Exponential Smoothing. *Euclid*, jurnal.ugj.ac.id, <http://jurnal.ugj.ac.id/index.php/Euclid/article/download/2194/1414>
- [4]. Vlandari, RT, & Parwitasari, TA (2018). Perbandingan Model AR (1), ARMA (1, 1), dan ARIMA (1, 1, 1) pada Prediksi Tinggi Muka Air Sungai Bengawan Solo pada Pos Pemantauan Jurug. *MUST: Journal of...*, journal.um-surabaya.ac.id, <http://journal.um-surabaya.ac.id/index.php/matematika/article/view/1620>
- [5]. Sean J. Taylor, Ben Letham (2017). *Prophet: Forecasting at Scale*. research.facebook.com, <https://research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/>
- [6]. Dahman W, Dadan. (2021). *Jenis-jenis Machine Learning*. Diakses pada 30 November 2021, dari <https://medium.com/sysinfo/jenis-jenis-machine-learning-50f918453336>
- [7]. Enes Zvorničanin. (2023). *Time Series Projects: Tools, Packages, and Libraries That Can Help*. Diakses pada 1 Februari 2023, dari

- <https://neptune.ai/blog/time-series-tools-packages-libraries>
- [8]. Prophet (2017). *Hyperparameter tuning*. facebook.github.io, <https://facebook.github.io/prophet/docs/diagnostics.html#hyperparameter-tuning>
- [9]. Assembled. *What is Short Term Forecasting*. <https://www.assembled.com/university/what-is-short-term-forecasting#:~:text=So%2C%20with%20that%20in%20mind,plan%20to%20meet%20those%20needs.>
- [10]. Bliemel, F (1973). *Theil's forecast accuracy coefficient: A clarification.*, journals.sagepub.com, <https://doi.org/10.1177/002224377301000413>
- [11]. Streamlit (2019). *A faster way to build and share data apps*. streamlit, <https://streamlit.io/>
- [12]. Treuille, Adrien (2019). *Turn Python Scripts into Beautiful ML Tools*. towardsdatascience, <https://towardsdatascience.com/coding-ml-tools-like-you-code-ml-models-ddba3357eace>
- [13]. Dicoding Intern (2021). *Apa itu GitHub? Berikut Cara Menggunakannya*. Dicoding, <https://www.dicoding.com/blog/apa-itu-github/>.
- [14]. GitHub (2008). *Let's build from here*. GitHub, <https://github.com/>.