

BAB II

TINJAUAN PUSTAKA

2.1 Time Series Forecasting

Peramalan atau disebut juga forecasting adalah suatu metode yang mempelajari tentang memperkirakan suatu nilai di masa depan berdasarkan data-data pada masa lalu [5]. Time Series Forecasting menjadi pendekatan yang digunakan untuk mendapatkan prediksi curah hujan Jakarta Barat. Time Series Forecasting merupakan sebuah bidang dalam Machine Learning yang berfokus pada atribut waktu dimana data yang didapat kemudian akan diolah menjadi prediksi untuk kejadian di masa depan. Time series forecasting adalah bidang penting dalam area forecasting atau peramalan di mana observasi masa lalu dari variabel yang sama dikumpulkan dan dianalisis untuk mengembangkan model yang menggambarkan hubungan dasar[6].

2.2 Interquartile Range (IQR)

IQR merupakan suatu ukuran statistik yang digunakan untuk menggambarkan sebaran atau penyebaran dari suatu set data. IQR mengukur rentang nilai tengah dari suatu data, yaitu rentang antara kuartil pertama (Q_1) dan kuartil ketiga (Q_3). Formula IQR adalah sebagai berikut:

$$IQR = Q_3 - Q_1 \quad (1)$$

Keterangan :

Q_3 = Jarak kuartil 3 data

Q_1 = Jarak kuartil 1 data

2.3 Moving Average

Simple Moving Average (SMA) digunakan untuk mengurangi fluktuasi dalam data dan memberikan gambaran yang lebih jelas tentang tren yang mendasarinya. Untuk setiap titik data dalam series, SMA dihitung dengan mengambil rata-rata aritmatika dari sejumlah titik data terbaru sesuai dengan panjang jendela yang ditentukan. Setelah rata-rata dihitung, titik rata-rata ini digunakan untuk membuat garis smooth (halus) yang memperlihatkan trend umum dalam data. SMA membantu mengurangi fluktuasi curah hujan dalam data, sehingga memperjelas tren dan pola yang lebih signifikan.

Formula SMA dijabarkan sebagai berikut:

$$SMA_t = \frac{x_t + x_{t-1} + x_{t-2} + \dots + x_{t-(n+1)}}{n} \quad (2)$$

Keterangan :

SMA_t = Nilai moving average pada waktu t

n = jumlah periode dalam moving average

x_t = Data pada periode t

2.4 Prophet

Prophet merupakan algoritma yang digunakan untuk melakukan prediksi data berbentuk *time-series* yang berdasar dari model aditif, dimana trend yang bersifat non-linear akan dicocokkan dalam deret waktu secara tahunan, mingguan, dan harian. *Prophet*, juga dikenal sebagai *Fbprophet*, adalah model peramalan deret waktu yang dikembangkan oleh Facebook Core Data Science Team [16]. Teknik Prophet didasarkan pada asumsi bahwa data deret waktu dapat dijelaskan sebagai

campuran dari berbagai karakteristik, seperti tren, musiman, dan hari libur [6]. model *Prophet* menerapkan model aditif umum atau General Adaptive Model (GAM) untuk memenuhi fungsi pemulusan (smoothing) dan peramalan.

Formula Prophet dapat dijabarkan sebagai berikut:

$$y_t = g(t) + s(t) + h(t) + e(t) \quad (3)$$

Keterangan :

$g(t)$ = Growth Term / Trend

$s(t)$ = Seasonal Pattern

$h(t)$ = Holiday Pattern

e_t = Error

Di sini, $g(t)$ adalah fungsi tren yang memodelkan perubahan non-periodik dalam nilai deret waktu, $s(t)$ mewakili perubahan periodik (misalnya, musiman mingguan dan tahunan), dan $h(t)$ mewakili efek libur yang terjadi dengan jadwal yang mungkin tidak teratur selama satu atau beberapa hari [21]. Model prophet menggunakan piecewise linear model memodelkan perubahan non-periodik, perubahan periodik, holiday pattern atau effect yaitu perubahan yang disebabkan karena adanya hari-hari tertentu, serta error yang disebabkan karena adanya perubahan tak terduga yang tidak dapat ditangkap oleh model.

Untuk mendapatkan nilai growth term atau trend pada kasus ini menggunakan Linear piecewise logistic growth model dengan persamaan:

$$g(t) = \frac{Ct}{1 + \exp(-(k+a(t)^T\delta)(t - (m + a(t)^T\delta)))} \quad (4)$$

Dimana

Ct = Time Varying Carrying Capacity

k = Growth Rate

m = offset Parameter

Untuk mendapat nilai seasonality pada Prophet, digunakan *Fourier series* pada model Prophet untuk menyediakan model yang fleksibel dari efek periodic. Persamaan untuk mendapatkan seasonal effect adalah sebagai berikut:

$$s(t) = \sum_{n=1}^N (a_n \cos(\frac{2\pi nt}{P}) + (b_n \sin(\frac{2\pi nt}{P}))) \quad (5)$$

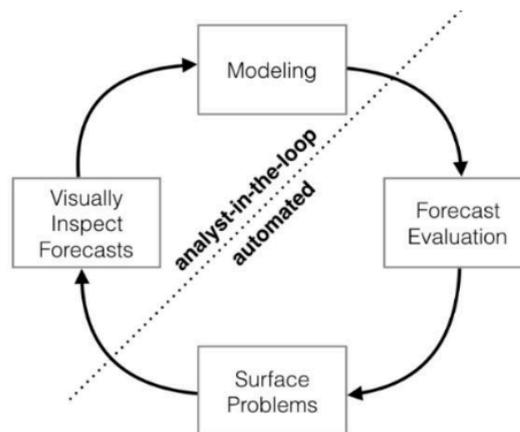
Dengan parameter $\beta = [a_1, b_1, \dots, a_n, b_n]^T$

Dimana

$$a_n = \frac{1}{P} \int_{-p}^p f(x) \cos(\frac{2\pi nt}{P}) dx$$

$$b_n = \frac{1}{P} \int_{-p}^p f(x) \sin(\frac{2\pi nt}{P}) dx$$

Untuk effect Holiday, dalam kasus curah hujan tidak perlu digunakan karena data curah hujan yang diambil tidak terpengaruh dengan adanya hari libur yang ada di Indonesia.



Gambar 2.4.1 Skema forecasting model Prophet

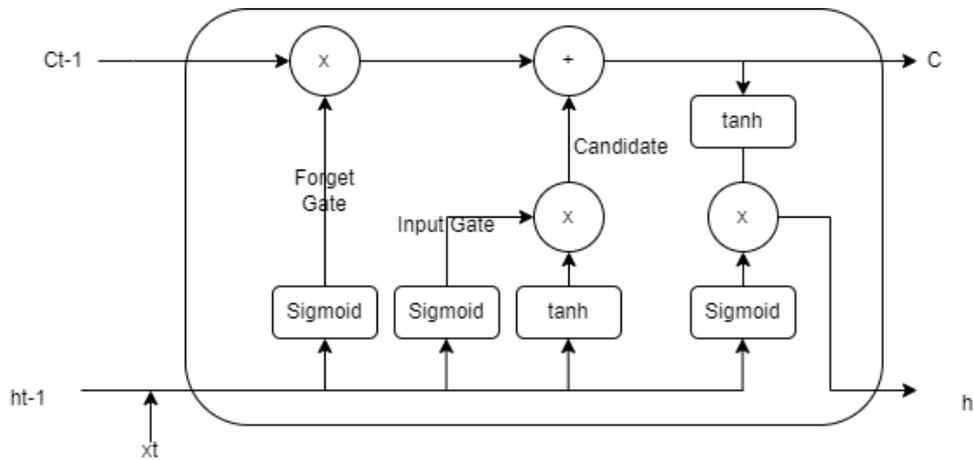
Algoritma Prophet merupakan algoritma semi-otomatis, yang masih membutuhkan peran manusia dalam melakukan pemodelan [10]. Pada model Prophet, proses forecasting diawali dengan tahap visualisasi data sebagai salah

satu ilustrasi proses pelatihan model. Lalu dilanjut ke proses training model dengan melakukan proses tuning parameter Trend, Seasonality, Holiday dan parameter komponen lainnya. Selanjutnya adalah evaluasi model forecasting. Apabila hasil evaluasi model sudah baik, dapat dilanjutkan ke tahap *Forecasting*, namun apabila hasil evaluasi kurang baik, dapat dilakukan modifikasi model untuk parameter yang bersangkutan sehingga hasil model yang didapat memiliki evaluasi yang baik. Proses terakhir berkaitan dengan *surface problems*, ketika terdapat masalah yang memerlukan partisipasi peneliti, maka peneliti dapat menganalisis hasil prediksi dan tidak ragu untuk memodifikasi model [10].

2.5 Long Short Term Memory

Long Short Term Memory (LSTM) merupakan salah satu bagian dari Recurrent Neural Network (RNN). LSTM mampu untuk belajar mengenai dependensi jangka panjang dan sangat populer untuk digunakan pada masalah data yang sequential, seperti data time series[17]. LSTM membahas mengenai masalah gradient yang hilang [18]. LSTM sangat cocok digunakan untuk menyelesaikan permasalahan berkaitan dengan klasifikasi, prosesing dan prediksi berdasarkan data time series. Hal tersebut dikarenakan terdapat kelangkaan durasi data yang tidak diketahui di antara barisan peristiwa penting dalam rangkaian waktu. LSTM akan memproses data yang masuk secara berulang dimana perulangan terjadi di setiap *output* dan *hidden state*, begitu seterusnya hingga hasil *output* yang diperoleh adalah *output* yang paling akurat. Informasi yang masuk ke *forget gate* nantinya akan diproses dengan informasi yang ada disimpan pada *memory cell*.

Proses LSTM Dapat dipaparkan sebagai berikut:



Gambar 2.5.1 Proses LSTM

Berdasarkan gambar 2.5.1, digunakan aktivasi sigmoid untuk mendapatkan nilai dari forget gate. Rumus menghitung nilai pada input gate dapat dipaparkan sebagai berikut:

$$f_t = \sigma(Wf_{ht} - 1 + Uf_{xt} + b_f) \quad (6)$$

Selanjutnya menggunakan rumus 7 dan 8 untuk menghitung nilai input gate pada memory cell. Hal tersebut dilakukan untuk mendapatkan serta memperbarui informasi dan menyimpan nilai informasi tersebut menggunakan fungsi aktivasi *tanh*.

Input Gate:

$$i_t = \sigma(Ui_{xt} + Wi_{ht} - 1 + b_i) \quad (7)$$

Intermediate cell state:

$$C_t = \tanh(Uc_{xt} + Wc_{ht} - 1 + b_c) \quad (8)$$

Selanjutnya menghitung nilai yang ada pada input gate menggunakan persamaan 9

Cell State

$$c_t = (f_t \times c_{t-1}) + (i_t \times C_t) \quad (9)$$

Kemudian *forget gate* akan mengisi nilai *memory cell* menggunakan *cell gate* yang akan dilanjutkan ke *output gate*. Pada *output gate*, ada 2 gate untuk menghitung nilai output, perhitungan menggunakan persamaan 10 dan 11.

Output Gate:

$$ot = \sigma(Woht - 1 + Uoht + bo) \quad (10)$$

Hidden State:

$$ht = ot \times Ct \quad (11)$$

keterangan:

xt = Vector input

U = Input-to-hidden Weight

W = hidden-to-hidden Weight

B = Bias Vector

2.6 Root Mean Square Error

Root Mean Square Error (RMSE) merupakan salah satu metode evaluasi yang sering digunakan untuk mengetahui performa model. Evaluasi RMSE dapat digunakan untuk mengetahui besarnya tingkat kesalahan dari hasil prediksi model, dimana semakin kecil (mendekati 0) nilai RMSE yang didapat maka hasil prediksi dari model akan semakin akurat. Dengan kata lain, RMSE seberapa baik model dapat meramalkan nilai sebenarnya, dan nilai RMSE yang semakin kecil menunjukkan kinerja model yang semakin baik. Untuk evaluasi, Root Mean Square Error (RMSE) banyak diadopsi dalam banyak sistem rekomendasi untuk mengukur perbedaan antara skor prediksi dan penilaian aktual pengguna [19]. Hasil pengujian dianggap baik jika nilai prediksi mendekati nilai target [20]. Nilai RMSE dapat dihitung dengan persamaan 12:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (12)$$

dimana :

n = jumlah data

y_i = nilai sebenarnya

\hat{y}_i = nilai prediksi

2.7 Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) adalah salah satu metrik yang digunakan untuk mengevaluasi kinerja model prediksi dalam analisis deret waktu atau time series. MAPE mengukur rata-rata kesalahan dalam bentuk persentase terhadap nilai aktual, yang memberikan gambaran seberapa besar error prediksi yang dilakukan oleh model relatif terhadap nilai yang sebenarnya. MAPE memiliki fitur-fitur penting dan diinginkan termasuk keandalan, ukuran bebas satuan, kemudahan interpretasi, kejelasan presentasi, dukungan evaluasi statistik, dan penggunaan semua informasi terkait kesalahan [22]. Semakin rendah persentase MAPE yang didapat menggambarkan bahwa model menunjukkan hasil forecasting yang semakin akurat.

Formula MAPE dijabarkan sebagai berikut:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (13)$$

Keterangan :

y_i = nilai aktual

\hat{y}_i = nilai prediksi

Sementara itu, Lewis (1982) menyusun sebuah tabel yang berisi nilai-nilai MAPE khas untuk data industri dan bisnis serta interpretasinya pada tabel 2.1 [22].

Tabel 2.1 Interpretasi nilai MAPE

MAPE	Interpretation
<10	Highly accurate forecasting
10-20	Good Forecasting
20-50	Reasonable Forecasting
>50	Inaccurate Forecasting
Source: Lewis (1982, p.40)	

