

BAB II

TINJAUAN PUSTAKA

2.1 FORECASTING

Teori peramalan didasarkan pada premis bahwa pengetahuan saat ini dan masa lalu dapat digunakan untuk membuat prediksi tentang masa depan. Ada keyakinan bahwa dengan mengidentifikasi pola dalam nilai historis akan berhasil memprediksi nilai masa depan. Sebaliknya, di antara banyak opsi untuk perkiraan satu deret waktu pada periode waktu mendatang adalah nilai yang diharapkan (dikenal sebagai perkiraan titik), interval prediksi, persentil, dan seluruh distribusi prediksi. Kumpulan hasil ini secara kolektif dapat dianggap sebagai "ramalan". [4]

Prosedur peramalan adalah yang terbaik jika berhubungan dengan masalah yang akan dipecahkan dalam praktik. Teori tersebut kemudian dapat dikembangkan dengan memahami fitur-fitur esensial dari masalah tersebut, hasil teoritis dapat mengarah pada peningkatan praktik. Dalam pendahuluan ini, diasumsikan bahwa teori peramalan dikembangkan sebagai metode dan model peramalan. Metode prakiraan didefinisikan di sini sebagai urutan langkah-langkah yang telah ditentukan sebelumnya yang menghasilkan prakiraan pada periode waktu mendatang. [4]

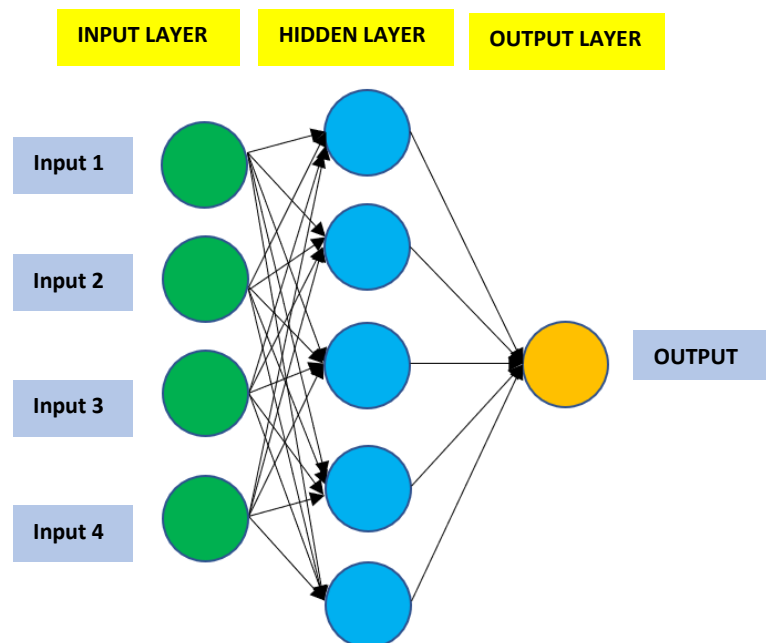
Peramalan harus menjadi bagian integral dari kegiatan pengambilan keputusan manajemen, karena dapat memainkan peran penting dalam banyak bidang perusahaan. Organisasi modern membutuhkan perkiraan jangka pendek, jangka menengah, dan jangka panjang, tergantung pada aplikasi spesifik (Hyndman & Athanasopoulos, 2018) [5] Peramalan berdasarkan horizontal waktu dicakup diklasifikasikan menjadi 3 kategori, antara lain (Heizer & Render, 2006) [6] :

1. Peramalan jangka pendek, peramalan ini mencakup jangka waktu hingga 12 bulan tetapi umumnya kurang dari tiga bulan.

2. Peramalan jangka menengah, peramalan ini pada umumnya mencakup waktu hingga 3 bulan hingga tiga tahun.
3. Peramalan jangka panjang umumnya peramalan ini untuk perencanaan 3 tahun atau lebih

2.2 JARINGAN SYARAF TIRUAN

Jaringan Syaraf Tiruan (JST) merupakan Sebuah mesin yang dirancang untuk memodelkan cara otak melakukan tugas atau fungsi tertentu yang menarik; jaringan biasanya diimplementasikan dengan menggunakan komponen elektronik atau disimulasikan dalam *software* pada komputer digital.[7]



Gambar 2. 1 Arsitektur Jaringan Syaraf Tiruan.

2.3 *EXTRAME LEARNING MACHINE (ELM)*

ELM adalah metode yang bekerja dengan konsep *single hidden layer feed forward networks* (SLFNs), metode ini diciptakan untuk mengatasi kelemahan metode jaringan syaraf tiruan feedforward lebih-lebih pada proses *learning speed*. ELM baik digunakan untuk mengatasi masalah peramalan dengan waktu pelatihan cepat, mudah untuk diaplikasikan pada masalah kompleks dan diterapkan pada kehidupan nyata. [8] Berdasarkan kelebihan yang dimiliki oleh ELM sehingga

fokus penelitian ini adalah mengukur performa kecepatan waktu komputasi dan seberapa akurat hasil yang didapatkan pada metode ELM. Nilai keakuratan akan dihitung dengan menggunakan *Mean Square Error* (MSE).[8]

Metode ELM memiliki beberapa tahapan dalam proses prediksiannya. Tahapan tersebut adalah proses normalisasi, *Training*, *Testing*, dan denormalisasi data. Pada penelitian ini, peneliti menggunakan metode *Mean Square Error* (MSE) untuk mencari tingkat kesalahan prediksi. [8]

Langkah-langkah perhitungan dengan metode ELM yaitu pertama adalah sebagai berikut :

2.3.1 Normalisasi Data

Normalisasi data dilakukan karena range nilai input tidak sama, yaitu bernilai puluhan hingga ribuan. *Input* akan diproses ke nilai *output* yang kecil sehingga data yang digunakan harus disesuaikan agar dapat diproses untuk mendapatkan nilai normalisasi yang kecil. Dalam penelitian ini, data yang digunakan akan disesuaikan dengan cara menormalisasi data. Berikut adalah proses normalisasi data menggunakan metode *Min-Max Normalization* (Jain and Bhandare 2011):

$$d^i = \frac{d - \min}{\max - \min}$$

Keterangan :

d' = nilai dari hasil normalisasi data

d = nilai asli data

min = nilai minimum pada data set fitur X

max = nilai maksimal pada data set fitur X

2.3.2 Proses *Training*

Proses *Training* harus dilalui sebelum melakukan proses prediksi. Tujuannya adalah untuk mendapatkan nilai *output weight*. Langkah-langkah proses *Training* yaitu sebagai berikut (Huang, Zhu and Siew, 2006) [9]:

1. Langkah pertama adalah menginisialisasi *input weight* dan bias. Nilai ini diinisialisasi secara acak dengan range antara -1 hingga 1.

2. Keluaran di *hidden layer* dihitung menggunakan fungsi aktivasi. Langkah pertama adalah menghitung keluaran *hidden layer* (H_{init}), setelah nilai H_{init} didapatkan kemudian dihitung menggunakan fungsi aktivasi sigmoid. Berikut adalah persamaan untuk menghitung keluaran di *hidden layer*:

$$H_{init\ ij} = (\sum_{k=1}^n W_{jk} \cdot X_{ik}) + b_j$$

Keterangan :

H_{init} = Matriks keluaran *hiddenlayer*.

i = [1,2,...,N], dimana N adalah keseluruhan jumlah data.

j = [1,2,..., \tilde{N}], dimana \tilde{N} adalah keseluruhan jumlah *hidden neuron*.

n = Jumlah *input neuron*.

w = Bobot *input*.

x = *Input* data yang digunakan.

b = Nilai bias.

3. Menghitung *output weight*. Untuk mendapatkan *output weight*, langkah pertama yang harus dilakukan adalah menstranspose matriks hasil keluaran *hidden layer* dengan fungsi aktivasi. Setelah ditranspose, matriks transpose tersebut dikalikan dengan matriks hasil keluaran *hidden layer* dengan fungsi aktivasi biasa disebut matriks H . Langkah selanjutnya adalah menghitung nilai invers dari matriks H tersebut. Setelah itu menghitung matriks *Moore-Penrose Generalized Invers* dari hasil keluaran *hidden layer* dengan fungsi aktivasi. Berikut adalah persamaan untuk menghitung nilai *output weight*:

$$\beta = H^+ T$$

Keterangan :

β = Matriks *Output weight*.

H^+ = Matriks *Moore-Penrose Generalized Invers* dari matriks H .

T = Matriks *Target*.

2.3.3 Proses *Testing*

Pada proses ini ini bertujuan untuk mengevaluasi metode ELM dari hasil proses *Training* sebelumnya. Proses *Testing* dilakukan menggunakan input weight, bias dan output weight yang didapatkan dari proses *Training*. Berikut langkah-langkah proses *Testing* adalah sebagai berikut:

1. Langkah pertama adalah menginisialisasi input weight dan bias yang telah didapatkan dari proses *Training*.
2. Keluaran di hidden layer dihitung menggunakan fungsi aktivasi. Pilih salah satu fungsi aktivasi yang digunakan yaitu fungsi aktivasi sigmoid, sin, hardlim.
3. Nilai output weight yang telah didapatkan pada proses *Training* digunakan untuk menghitung keluaran output layer yang merupakan hasil prediksi. Berikut adalah persamaan untuk menghitung nilai *outputlayer*:

$$y = H\beta$$

Keterangan :

y = *Output layer* yang merupakan hasil prediksi.

β = nilai *output weight* didapatkan dari proses *Training*.

H = Keluaran di *hidden layer* dihitung dengan fungsi aktivasi.

4. Langkah terakhir adalah menghitung nilai *error* semua output layer. Nilai *error* ini menunjukkan nilai kesalahan dari hasil prediksi yang didapatkan.

2.3.4 Mean Square Error (MSE)

Mean Square Error (MSE) digunakan untuk mengevaluasi hasil prediksi. Berikut adalah persamaan untuk menghitung nilai *error* pada hasil prediksi (Makridakis and Hibon, 1995):

$$MSE = \frac{\sum_{i=1}^n e_i^2}{n} = \frac{\sum_{i=1}^n (y_i - t_i)^2}{n}$$

Keterangan :

n = Jumlah data.

e = *Error*.

y_i = Nilai output (prediksi).

t_i = Nilai aktual.

2.3.5 Proses Denormalisasi Data

Proses ini berfungsi untuk membangkitkan nilai yang telah dinormalisasi menjadi nilai asli. Berikut adalah persamaan untuk proses denormalisasi data:

$$d = d' (\max - \min) + \min$$

Keterangan :

d' = nilai hasil prediksi sebelum didenormalisasi

d = nilai asli setelah didenormalisasi

min = nilai minimum pada data set fitur X

max = nilai maksimal pada data set fitur X

2.4 BAHASA PEMROGRAMAN PYTHON

Python adalah bahasa pemrograman tingkat tinggi yang ditafsirkan, berorientasi objek, dengan semantik dinamis. Pemrograman tingkat tinggi yang dibangun dalam struktur data, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk pengembangan aplikasi secara cepat, serta digunakan sebagai bahasa scripting untuk menghubungkan komponen yang ada bersama-sama. Sintaksis *Python* yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program.[10]

Python mendukung modul dan paket, yang mendorong modularitas program dan penggunaan kembali kode. *Interpreter* Python dan pustaka standar yang luas tersedia dalam bentuk sumber atau biner tanpa biaya untuk semua platform utama, dan dapat didistribusikan secara bebas. Python memiliki banyak library yang baik untuk proses data science. Python mempunyai library Numpy yang berfungsi memudahkan proses komputasi ilmiah seperti data kuantitatif. Sedangkan untuk memvisualisasikannya agar mudah dipahami dan lebih atraktif, disediakan library Matplotlib.

2.4.1 Library Python

1. Numpy : Guido van Rossum, Jim Fulton dan Jim Hugunin merancang implementasi matriks untuk operasi numerik yang disebut Numeric kemudian

mengubah namanya menjadi Numpy. Python memiliki struktur data daftar, tetapi hal itu tidak cukup untuk melakukan konversi dan perhitungan numerik. Jadi Numpy pada dasarnya dikembangkan pada struktur data inti yang disebut ndarray. ndarray adalah jenis array matriks, yang memiliki elemen jenis yang sama. Array numpy secara default memiliki ukuran dan bentuk tetap $m \times n$ dengan m baris dan n kolom yang sama ketika elemen baru ditambahkan ke matriks melebihi ukurannya, kemudian menyalin semua elemen dari array yang ada saat itu dan membuat array baru dengan ukuran yang sama size kemudian menghapus array asli. Numpy dapat digunakan di lingkungan pengembangan terintegrasi python apa pun. Misalnya: Inisialisasi: `import numpy sebagai np` Perintah ini mengimpor perpustakaan numpy dengan objek np.

Syntax untuk membuat larik: `a = np.array([1, 2, 3])` Kode ini membuat objek ndarray dan menyisipkan 1, 2, 3 angka ke variabel a.

2. Pandas : Pandas library adalah alat yang paling efisien untuk pemrosesan data dan operasi manipulasi, rangkaian struktur data dan bingkai datanya bertindak sebagai blok bangunan untuk analisis data. Seri adalah array satu dimensi yang sama dengan daftar di python sedangkan bingkai data adalah array multidimensi. Dengan menggunakan panda, kita dapat memuat dan membaca semua jenis data seperti file Json, Ms excel.

Syntax inisialisasi: `import panda sebagai pd` Perintah ini mengimpor perpustakaan pandas dengan objek pd. Membaca file: `Df=pd.read("nama file")`

Fungsi baca di perpustakaan pandas digunakan untuk memuat semua file ke dalam bingkai data. Pustaka Pandas melakukan banyak fungsi seperti mengiris, menggabungkan, mengelompokkan.

3. Matplotlib: matplotlib digunakan untuk visualisasi data. Library Ini menyediakan berbagai macam fungsi, metode dan grafik visual untuk memplot titik data pada grafik dan untuk menganalisis. Library ini menyediakan berbagai grafik untuk mengeksplorasi data seperti plot garis, beberapa sub plot, gambar, kontur dan warna semu, histogram, jalur, plot tiga dimensi, plot aliran, dll., Dapat diintegrasikan di berbagai platform dan perangkat lunak untuk

visualisasi data. Ini memberikan kontrol penuh atas representasi data. Setelah diinisialisasi dengan sumbu x dan koordinat sumbu y maka setiap data yang rumit dapat direpresentasikan dalam grafik untuk mengetahui wawasannya.

Inisialisasi: Impor matplotlib.pyplot sebagai plt Kode ini mengimpor pustaka matplotlib ke lembar kerja.

4. Skit-learn : Perpustakaan ini dikembangkan untuk berfungsi dengan perpustakaan scipy dan Numpy untuk merancang dan mengimplementasikan model. library ini memiliki berbagai algoritma dari kombinasi statistik dan probabilitas. Dengan menggunakan pustaka ini, kumpulan data asli dapat dibagi menjadi kumpulan data latih dan uji yang diinginkan. Pengembang melatih model dari dataset pelatihan yang sebelumnya dibagi dari dataset asli dan kemudian mengevaluasinya dari dataset uji untuk mengukur akurasi, kualitas model. Pustaka Skit-learn juga menyediakan berbagai metode untuk supervised, unsupervised dan penguatan sedang belajar. Klasifikasi dan pengelompokan item data dapat dilakukan dengan menggunakan library ini.
5. Scipy : Python ilmiah yang disebut scipy digunakan untuk manipulasi array dimensi-N. Pustaka ini berjalan pada inti numpy ndarray. Perpustakaan ini menyediakan banyak metode untuk perhitungan ilmiah seperti optimasi, pemrograman linier, menghitung jarak. [10]

2.5 TABLEU

Tableau merupakan salah satu alat visualisasi untuk mengelola data secara model grafik yang baik. Dengan memanfaatkan visualisasi ini Visualisasi grafik dari representasi data jauh lebih kuat daripada tampilan dalam bentuk angka. Sementara tabel memerlukan kita untuk membaca dan mempertimbangkan makna dan hubungan setiap nilai individu yang disajikan, visualisasi mengizinkan untuk memproses banyak nilai-nilai secara bersamaan. Lebih lanjut, manusia dapat lebih efisien dan efektif proses grafik dari tabel numerik teks. Dengan demikian, visualisasi memungkinkan analisis untuk mengenali tren, tempat pola, dan mengidentifikasi dengan cepat dan optimal. [11]