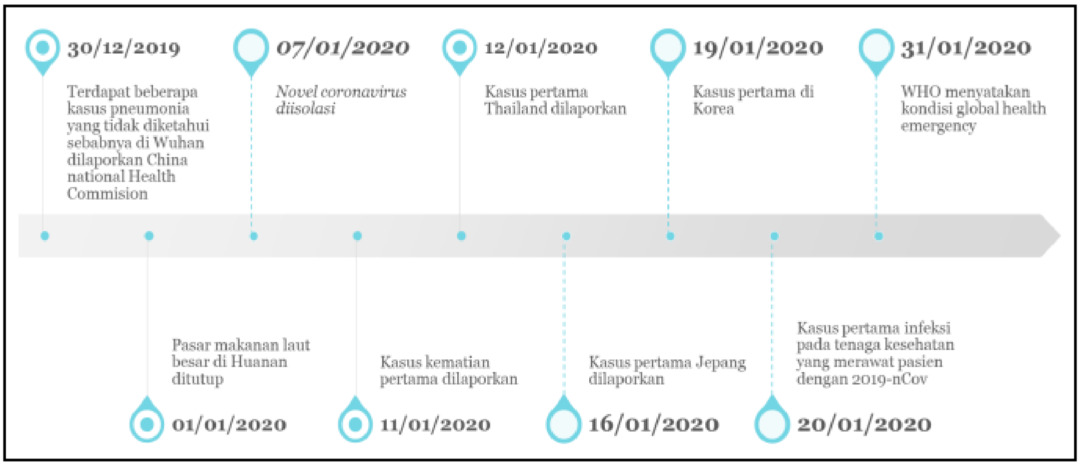
BAB 2

LANDASAN TEORI

2.1. Virus Corona (nCoV-2019)

Virus corona 2019 atau nCoV-2019 merupakan sebuah virus yang muncul pertama kali di Kota Wuhan, Provinsi Hubei, China pada awal Desember 2019. Kemunculan virus ini pertama kali diketahui setelah pemerintah China melaporkan adanya 44 pasien dengan gejala *pneumonia* berat kepada Badan Kesehatan Dunia (WHO). Dugaan awal terkait kejadian ini adalah karena adanya transaksi jual beli hewan secara ilegal di Pasar Hewan Wuhan. Pada 10 Januari 2020, penyebab kasus ini mulai teridentifikasi sebagai varian virus corona baru (Handayani *et al.*, 2020). Tidak lama setelahnya, terdapat laporan dari beberapa negara terkait kemunculan varian virus baru ini di wilayahnya. Beberapa negara tersebut antara lain Jepang, Korea Selatan, Perancis, Thailand, hingga total 25 negara lainnya.



Gambar 2.1. Alur Waktu Kejadian Virus Corona (Handayani *et al.*, 2020)

Virus corona merupakan virus terbesar dalam ordo *Nidovirales*, dan termasuk dalam familia *Coronaviridae*, serta merupakan genus *Betacoronavirus*. Virus ini berbentuk bulat dengan diameter sekitar 125 mm. Organisasi Kesehatan Dunia (WHO) mengungkapkan penyebab penyakit Covid-19 adalah *severe acute respiratory syndrome coronavirus 2* (SARS-CoV-2). Hal ini dikarenakan virus Covid-19 memiliki kesamaan struktur dengan SARS-CoV yang sempat menjadi pandemi juga di dunia (Parwanto, 2020).

Hasil penelitian oleh Institute of Virology di Wuhan mengidentifikasi virus ini sebagai varian baru yang berpotensi berbahaya dan menyebutnya *novel coronavirus 2019* (nCoV-2019). Sekarang penyakit ini populer dengan nama *coronavirus disease-19* atau Covid-19 (Parwanto, 2020). Sampai Januari 2021, virus ini telah menjangkit lebih dari 100 juta kasus di seluruh dunia (Worldometer, 2021).

2.2. Analisis Regresi

Analisis Regresi merupakan suatu metode yang digunakan untuk mengetahui hubungan antara suatu variabel dependen terhadap satu atau lebih variabel independen (Ghambir *et al.*, 2020). Salah satu model regresi paling umum adalah regresi linear seperti pada Rumus 2.1.

$$\mathrm{y} = \alpha + \beta X + \varepsilon \tag{2.1}$$

dimana :

y = variabel dependen (variabel yang akan coba diprediksi)

X = variabel independen (variabel yang digunakan untuk memprediksi y)

α = konstanta

β = nilai koefisien regresi (kemiringan)

ε = nilai *error* (perbedaan nilai prediksi y dengan nilai y yang sebenarnya)

Model lain dari regresi adalah regresi polynomial seperti pada rumus 2.2. Regresi polynomial sering digunakan pada *range* data lebar dan tidak berbentuk linear. Regresi polynomial dapat memperkirakan hubungan antara variabel dependen dan independen dengan akurat dan mudah dikombinasikan dengan fungsi lain.

$$y = \alpha_0 + \alpha_1 * X_1 + \alpha_2 * X_2^2 + .. + \alpha_n * X_n^n \quad (2.2)$$

dimana :

y = variabel dependen (variabel yang akan coba diprediksi)

X = variabel independen (variabel yang digunakan untuk memprediksi *y*)

α = nilai koefisien regresi untuk data ke - *n*

2.3. K-Nearest Neighbors

K-Nearest Neighbors merupakan suatu metode klasifikasi yang digunakan untuk mengelompokkan data baru berdasarkan jarak terhadap kumpulan data atau tetangga terdekat. Langkah-langkah penggunaan algoritma K-Nearest Neighbors adalah sebagai berikut.

1. Menentukan jumlah tetangga terdekat

   Pertama harus ditentukan jumlah *k* tetangga yang akan digunakan. Optimalnya jumlah *k* berada diantara 5 sampai 10. Jumlah *k* yang akan dipakai ditentukan dengan jumlah data yang ada.

2. Menghitung jarak antar poin data

Penghitungan jarak antar 2 data dapat dilakukan menggunakan *Euclidean distance* seperti pada rumus 2.3.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.3}$$

dimana :

d(p, q) = jarak antar data

x_i = koordinat atau nilai awal data ke - *i*

y_i = koordinat atau nilai akhir data ke - *i*

n = dimensi atau jumlah data

3. Menentukan kategori objek baru

   Setelah menghitung jarak antar data, kemudian akan diambil *k* tetangga terdekat sesuai kalkulasi jarak yang didapat sebelumnya. Pada *k* tetangga ini, akan dihitung jumlah titik data di setiap kategori. Terakhir, data baru akan dimasukkan ke kategori yang *k* tetangganya paling banyak.

2.4. K-Nearest Neighbors Regression

Salah satu metode KNN adalah K-Nearest Neighbors Regression. Metode ini digunakan untuk menganalisis nilai prediksi berdasarkan hubungan antara variabel dependen dan independen (Xiao, Ma and Ding, 2018). Pemodelan akan dibagi menjadi data training dan data uji. Jumlah data training akan dibuat lebih banyak karena digunakan untuk melatih algoritma, sedangkan data testing digunakan untuk mengetahui performa algoritma yang telah dilatih menggunakan data training. Pada metode ini, data *training* akan ditempatkan pada suatu variabel bernama X dimana X = {Xi, i = 1, …, n}. Pada sampel *training*, akan dicari *k* sampel yang memiliki jarak terdekat dengan sampel uji z, lalu akan diprediksi nilai sampel ujinya berdasarkan k sampel ini. Untuk pengukuran jaraknya akan digunakan *Euclidean distance* seperti pada rumus 2.3. Dari nilai k sampel yang didapatkan sebelumnya, kita dapat menghitung prediksi nilai (Y_Z) untuk sampel uji z. Metode yang digunakan untuk menghitung estimasi nilai ini adalah *mean method* seperti pada rumus 2.4 (Khelifi and Jiang, 2011). Alur kerja K-Nearest Neighbors Regression dapat dilihat lebih detail pada gambar 2.2.

$$y_z = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (2.4)$$

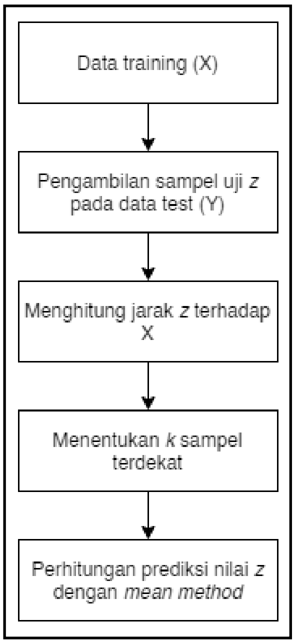
dimana :

y_z = nilai prediksi *mean method*

n = jumlah tetangga terdekat

y_i = nilai data tetangga terdekat ke – *i*

Dengan menggunakan *mean method*, akan didapatkan rata-rata sampel uji z setelah ditambahkannya *k* sampel yang sebelumnya didapatkan.



Gambar 2.2. Alur Kerja KNN Regression (Xiao, Ma and Ding, 2018)

2.5. Perhitungan Akurasi dan Error Rate

Pada penelitian ini akan digunakan lima metode evaluasi, yaitu *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), *Percentage Prediction Error* (PE), *Mean Absolute Percentage Error* (MAPE), dan *R^2 Metrics*.

2.5.1. Mean Absolute Error (MAE)

MAE adalah rata-rata perbedaan absolut antara prediksi dengan nilai actual (Brownlee, 2016). Rumus MAE seperti pada rumus 2.5.

$$\varepsilon_{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\, \hat{y}_i - y_i \,| \tag{2.5}$$

dimana :

n = jumlah data

$\hat{y}_i$ = nilai prediksi untuk data ke - *i*

y_i = nilai aktual untuk data ke - *i*

2.5.2. Mean Squared Error (MSE)

MSE adalah metode untuk menghitung *error rate* yang hampir sama dengan MAE, perbedaannya yaitu rata-rata nilai perbedaan antara prediksi dan nilai aktual akan dikuadratkan. Metode ini disebut juga dengan *Root Mean Squared Error* (RMSE) (Brownlee, 2016). Rumus MSE seperti pada rumus 2.6.

$$\varepsilon_{MSE} = \frac{1}{n}\sum_{i=1}^{n} (\, \hat{y}_i - y_i \,)^2 \tag{2.6}$$

dimana :

n = jumlah data

$\hat{y}_i$ = nilai prediksi untuk data ke - *i*

y_i = nilai aktual untuk data ke - *i*

2.5.3. Percentage Prediction Error (PE)

PE adalah metode untuk menghitung persen kesalahan antara nilai prediksi dengan nilai aktual. PE dapat digunakan untuk mengetahui tingkat kesalahan prediksi setiap data, maupun secara keseluruhan. Penggunaan PE seperti pada rumus 2.7.

$$PE = \left| \frac{nilai\ aktual - nilai\ prediksi}{nilai\ aktual} \right| x\ 100 \qquad (2.7)$$

2.5.4. Mean Absolute Percentage Error (MAPE)

MAPE adalah nilai rata-rata dari persentase kesalahan absolut sebuah hasil prediksi. Metode ini digunakan untuk mengetahui seberapa akurat sebuah hasil *forecast* / perkiraan secara keseluruhan pada sebuah sistem. Rumus MAPE seperti pada rumus 2.8.

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| x\ 100 \qquad (2.8)$$

dimana :

n = jumlah data

$\hat{y}_i$ = nilai prediksi untuk data ke - *i*

y_i = nilai aktual untuk data ke - *i*

2.5.5. R^2 Metrics

Metrik R^2 merupakan metode untuk menghitung nilai hubungan antara variabel dependen (X) dan variabel independen (Y). Metode ini juga sering disebut sebagai koefisien determinasi. Nilai hubungan yang didapat akan berada pada rentan 0 sampai 1 (Brownlee, 2016). Rumus R^2 seperti pada rumus 2.9.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \qquad (2.9)$$

dimana :

$\hat{y}_i$ = nilai prediksi untuk data ke - *i*

y_i = nilai aktual untuk data ke - *i*

$\bar{Y}$ = nilai rata-rata prediksi

2.6. Min-Max Scaling

Data Scaling merupakan sebuah metode yang digunakan untuk membuat data numerik memiliki rentang nilai yang sama pada suatu *dataset*. Salah satu jenis *data scaling* adalah *min-max scaling*. Metode ini akan mengubah nilai data menjadi rentang nilai 0 sampai 1. Nilai kosong akan berarti nilai minimum variabel sedangkan nilai satu berarti nilai maksimum variabel. *Min-max scaling* cocok digunakan untuk berbagai model pembelajaran mesin, terutama apabila nilai data cukup besar dan memiliki *range* yang lebar. Cara kerja *min-max scaling* adalah setiap nilai pada sebuah fitur akan dikurangi dengan nilai minimum dari fitur tersebut, kemudian akan dibagi dengan nilai maksimum fitur dikurangi nilai minimum fitur tersebut. Rumus *min-max scaling* dapat dilihat pada rumus 2.10 sebagai berikut.

$$X_{new} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}} \tag{2.10}$$

dimana :

X_{new} = nilai data baru

X_{old} = nilai data lama

X_{max} = nilai data tertinggi dari fitur

X_{min} = nilai data terendah dari fitur