

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Pencemaran Udara**

Pencemaran udara adalah udara bersih yang terkontaminasi zat pencemar di udara yang dapat mengganggu kesehatan manusia, menghasilkan dampak berbahaya bagi ekosistem dan dapat mengakibatkan perubahan iklim (Valerro, 2008). Sejumlah zat yang dilepaskan ke udara oleh aktivitas industri dan kegiatan lainnya yang dapat membahayakan kesehatan manusia, hewan dan tumbuhan disebut pencemaran udara. Udara bersih adalah udara yang tidak berbau, tidak berwarna serta kualitasnya layak untuk pernapasan. Pencemaran udara setiap hari secara terus-menerus dilepaskan ke udara dari aktivitas manusia, industri dan kendaraan bermotor. Adapun karena sifat alami udara yang bisa menyebar tanpa ada batasan ruang, membuat dampak pencemaran udara bisa bersifat lokal, kawasan maupun global. Udara yang tercemar seringkali tidak tertangkap oleh panca indra kita dan potensi bahayanya tetap ada. Salah satu bentuk pencemaran udara yang dapat dilihat yaitu gas buangan hasil pembakaran yang berbentuk asap tebal hitam yang berarti komponen partikel di dalam asap tersebut sangat banyak (Laridis, 2010).

Pencemaran udara semakin memburuk seiring dengan kemajuan teknologi, dimana dengan adanya kemajuan teknologi sehingga sumber penghasil polusi udara semakin meningkat. Hal ini dapat dilihat dari semakin banyaknya pabrik-

pabrik, industri, kendaraan bermotor dan pembangkit listrik yang setiap harinya menyumbang emisi polusi udara. Dari pencemaran ini, sangat banyak pihak yang dirugikan akibat udara yang tidak bersih untuk pernapasan, ketidaknyamanan, gangguan pada kesehatan, kerusakan lingkungan dan juga sebagai pemicu pemanasan global.

Ditinjau dari sifat pencemar udara terdapat 3 jenis yaitu sumber titik (*point source*), sumber kawasan (*area source*), dan sumber garis (*line source*). Sumber titik adalah sumber pencemar yang berasal dari sumber stasioner dan dapat diidentifikasi keberadaannya, misalnya emisi gas dari cerobong pabrik. Sumber kawasan adalah sumber pencemar yang berasal dari beberapa sumber titik yang berdekatan atau berasal dari sumber yang bergerak, misalnya pencemaran yang diakibatkan kepadatan lalu lintas. Sumber garis adalah sumber yang berasal dari emisi yang berbentuk garis, misalnya gas buang dari mobil atau sepeda motor ketika bergerak.

Berdasarkan karakteristik fisiknya, zat pencemar dibedakan atas partikular (aerosol) dan zat pencemar dalam bentuk gas. Partikular maupun zat pencemar dalam bentuk gas dapat tersusun dari senyawa anorganik dan organik. Beberapa zat yang terdapat pada polutan yang mengakibatkan pencemaran udara diantaranya karbon monoksida, karbon dioksida, nitrogen dioksida, sulfur dioksida, hidrokarbon, chlorofluorocarbon, timbal dan partikular. Zat-zat tersebut dihasilkan dari aktivitas manusia, salah satu contohnya adalah gas sulfur dioksida yang dihasilkan dari hasil aktivitas pembakaran industri dan pembangkit listrik

yang berbahan bakar batubara. Sulfur dioksida adalah gas berbau tajam, tidak berwarna dan tidak bersifat korosi hasil dari pembakaran bahan bakar batubara.

Menurut Miller & Spoolman (2009), terdapat beberapa polutan yang cukup dominan di udara yaitu:

1. Karbon oksida terdiri dari karbon monoksida dan karbon dioksida. Karbon monoksida merupakan suatu gas beracun, tidak berbau, dan tidak berwarna yang dihasilkan akibat pembakaran tidak sempurna dari material karbon. Sementara itu, karbon dioksida merupakan gas tidak berbau dan berwarna yang sebagian besar dihasilkan oleh siklus karbon alami.
2. Nitrogen oksida terdiri dari nitrogen dioksida dan nitrogen monoksida. Nitrogen monoksida terbentuk dari rekasi oksigen dengan nitrogen di udara. Sementara itu, nitrogen dioksida terbentuk akibat reaksi nitrogen monoksida dengan oksigen. Nitrogen monoksida tidak berwarna dan nitrogen dioksida berwarna cokelat kemerahan.
3. Sulfur dioksida merupakan gas yang tidak berwarna namun memiliki bau yang menyengat yang sebagian besar berasal dari sumber alami.
4. Partikulat tersuspensi terdiri dari berbagai macam partikel padat dan tetesan cairan yang dapat tetap tersuspensi di udara dalam waktu yang lama karena sangat kecil dan ringan. Partikulat tersuspensi memiliki dua jenis yang sangat membahayakan yaitu partikel halus (berukuran kurang dari 10  $\mu\text{m}$  atau PM10) dan partikel sangat halus (berukuran kurang dari 5  $\mu\text{m}$  atau PM2,5).

5. Ozon yaitu gas yang sangat reaktif dan tidak berwarna. Ozon sebagai polutan terletak di troposfer sementara ozon yang berada di stratosfer berfungsi sebagai pelindung makhluk hidup dari paparan radiasi sinar matahari.
6. Volatile Organic Compounds (VOCs) merupakan gas-gas organik di atmosfer seperti C<sub>3</sub>H<sub>8</sub>, C<sub>10</sub>H<sub>15</sub>, dan CH<sub>4</sub>.

## 2.2. *Particulate Matter 10 µm (PM10)*

Partikulat tersuspensi adalah partikulat yang terbentuk dari berbagai macam partikel solid dan tetesan cairan yang sangat kecil dan ringan untuk tetap tersuspensi di udara dalam waktu yang lama (Miller & Spoolman, 2009). Sementara itu, berdasarkan Vallero (2008) bahwa partikulat yang berada di atmosfer dapat dihasilkan langsung dari sumber maupun reaksi kimia yang terjadi antara uap air dengan gas. Umumnya partikulat dibagi menjadi dua kategori berdasarkan beratnya yaitu partikulat yang berukuran  $\leq 2.5 \mu\text{m}$  dan  $\leq 10 \mu\text{m}$ . Menurut Miller & Spoolman (2009), partikulat yang berukuran kurang dari 2,5  $\mu\text{m}$  (PM<sub>2,5</sub>) atau dapat disebut partikel sangat halus merupakan salah satu dari partikulat tersuspensi yang sangat berbahaya selain partikel halus atau partikulat berukuran kurang dari 10  $\mu\text{m}$  (PM<sub>10</sub>).

Partikel PM<sub>10</sub> terdiri dari partikel kompleks berukuran 0,1  $\mu\text{m}$ –10  $\mu\text{m}$ , mencakup semua ukuran virus (0,1  $\mu\text{m}$ –1  $\mu\text{m}$ ) dan bakteri (0,5  $\mu\text{m}$ –5  $\mu\text{m}$ ) (Lai *et al.*, 2009). PM<sub>10</sub> merupakan salah satu oksidan pencemar yang dapat dihisap oleh saluran pernapasan (Yusnabeti *et al.*, 2010). Konsentrasi debu di

atmosfer didominasi oleh PM10. Hrdličková *et al.* (2008) menyebutkan bahwa faktor yang mempengaruhi konsentrasi partikel debu (PM10) adalah suhu, kelembaban, kecepatan angin, dan arah angin. Perbedaan tempat atau heterogenitas secara spasial juga berpengaruh pada konsentrasi partikel debu (PM10) (Chu *et al.*, 2015).

Ukuran aerodinamik partikulat seperti PM10 tidak hanya menentukan kemudahan masuknya ke dalam paru-paru, tetapi juga sangat besar kemungkinannya memasuki lebih dalam lagi dari bagian paru-paru. Oleh karena itu, dalam waktu singkat, pajanan PM10 dapat menimbulkan gangguan saluran pernapasan/gejala Infeksi Saluran Pernapasan Akut (ISPA) (WHO, 2005). Konsentrasi PM10 yang tinggi dapat menyebabkan gangguan kesehatan terutama pada manusia seperti efek akut (batuk, sesak napas, nyeri dada, iritasi pada mata, detak jantung tidak beraturan, menurunkan fungsi paru, memperparah penyakit paru dan jantung) dan efek kronis (gangguan pada sistem saraf dan pembuluh darah dan memicu kematian dini pada orang dengan penyakit paru dan jantung) (Wulandari, 2016:680).

Antisipasi untuk menghindari dampak-dampak tersebut salah satunya dengan menerapkan batas konsentrasi PM10 yang diperbolehkan. Indonesia memiliki ambang batas yang diperbolehkan untuk PM10 atau dapat disebut baku mutu. Baku mutu untuk PM10 sebagaimana yang diatur dalam Peraturan Pemerintah Republik Indonesia Nomor 41 Tahun 1999 adalah 65  $\mu\text{g}/\text{Nm}^3$  untuk waktu pengukuran 24 jam dan 15  $\mu\text{g}/\text{Nm}^3$  untuk waktu pengukuran 1 tahun. Selain Indonesia, Amerika Serikat melalui *Environmental Protection Agency*

(EPA) menetapkan standar baku mutu yang berlaku di negaranya. EPA menetapkan baku mutu untuk PM10 yaitu 25  $\mu\text{g}/\text{m}^3$  dengan waktu pengukuran 24 jam dan 10  $\mu\text{g}/\text{m}^3$  dengan waktu pengukuran 1 tahun (Mines, 2014). Sementara itu World Health Organization (WHO) menetapkan baku mutu PM10 yang terbaru pada tahun 2005, yaitu 20  $\mu\text{g}/\text{m}^3$  dengan waktu pengukuran 24 jam dan 10  $\mu\text{g}/\text{m}^3$  dengan waktu pengukuran 1 tahun (WHO, 2006).

### 2.3. *Time Series*

*Time Series* (Deret Waktu) adalah serangkaian nilai-nilai variabel yang disusun berdasarkan waktu. Analisis *time series* mempelajari pola gerakan nilai-nilai variabel pada satu interval waktu seperti mingguan, bulanan, tahunan yang teratur. Metode *time series* didasarkan pada asumsi bahwa pola lama akan terulang. Manfaat dari analisis *time series* dapat diperoleh ukuran-ukuran yang dapat digunakan untuk membuat keputusan pada saat ini, untuk peramalan dan untuk merencanakan masa depan (Jayanti, 2013)

Data yang masuk dalam jenis *time series* dapat di lihat plot berdasarkan waktu. Hal ini dilakukan untuk mengamati pola dari data untuk selanjutnya menentukan langkah analisis yang akan dilakukan. Berdasarkan bentuk pola yang dibentuk, data *time series* dapat dibagi menjadi empat, pada umumnya, suatu data dapat terdiri atas satu atau beberapa komponen dari empat komponen utama yaitu (Hansun, 2016):

1. *Trend* (T)

Trend adalah komponen jangka panjang yang menunjukkan kenaikan atau penurunan dalam data *time series* untuk suatu periode waktu tertentu. Dengan lebih sederhana, dapat dikatakan bahwa trend adalah suatu garis atau kurva yang menunjukkan suatu kecenderungan umum dari suatu data runtun waktu.

2. Siklus (*Cycle/C*)

Komponen siklus adalah deret yang tidak beraturan berupa fluktuasi gelombang atau siklus dengan durasi waktu yang panjang. Komponen ini biasanya berhubungan dengan siklus bisnis (*business cycle*), suatu gerakan dianggap sebagai siklus apabila timbul kembali setelah jangka waktu lebih dari satu tahun.

3. Musiman (*Seasonality/S*)

Komponen musiman adalah suatu pola fluktuasi permintaan (*demand*) di atas atau di bawah garis trend yang terjadi tiap tahunnya. Fluktuasi musiman yang dimaksud dapat diklasifikasikan secara kuartal, bulanan, mingguan, atau harian, dan mengarah pada pola yang berubah secara regular dalam suatu waktu.

4. *Irregular (I)*

Komponen *irregular* adalah gerakan fluktuasi yang diakibatkan oleh kejadian yang tidak dapat diprediksi atau kejadian non-periodik.

#### 2.4. *Artificial Neural Network* (ANN)

*Artificial Neural Networks* (ANN) atau Jaringan Saraf Tiruan (JST) merupakan suatu model kecerdasan yang diilhami dari struktur otak manusia yang kemudian di implementasikan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran berlangsung. ANN yaitu suatu model penalaran yang didasarkan pada otak manusia. ANN terdiri dari sejumlah prosesor sangat sederhana dan saling berhubungan yang disebut neuron (Kurniawansyah, 2018). Neuron memiliki karakteristik yang sama dalam ANN, terdiri dalam kelompok-kelompok yang disebut layer. Neuron-neuron yang ada dalam satu layer terhubung dalam layer-layer lainnya yang berdekatan. Kekuatan hubungan antara neuron yang berdekatan dipresentasikan dalam kekuatan hubungan atau bobot. (Dharma, Putera, & Ardana, 2011).

Langkah penting dalam pengembangan sebuah model ANN yaitu penentuan bobot matrik melalui pelatihan atau *training*. Ada dua tipe mekanisme *training* yaitu *supervised training* dan *unsupervised training*, *Supervised training* memerlukan supervisi dari luar untuk memandu proses *training*. Algoritma ini menggunakan sejumlah pasangan data input-output yang dipergunakan sebagai contoh, dimana data yang dipergunakan sebagai contoh sebaiknya menggunakan data yang sudah diketahui kebenarannya. Output dari jaringan lalu dibandingkan dengan data output yang diharapkan (output contoh) untuk mendapatkan selisih dari output perkiraan dengan output sebenarnya. Selisih inilah yang dipergunakan untuk mengubah bobot jaringan sehingga diperoleh output yang sama atau output

yang mendekati target. Mekanisme sebuah *supervised training* disebut algoritma *backpropagation training* pada umumnya digunakan dalam aplikasi-aplikasi *engineering*. Karena pada ANN tidak mempertimbangkan fisik dari permasalahan, ANN merupakan sebuah model *blackbox* namun dapat mendeteksi proses fisik dalam model ANN yang sudah ditraining. (Dharma, Putera, & Ardana, 2011).

## 2.5. Recurrent Neural Network (RNN)

RNN adalah sejenis jaringan saraf tiruan yang memiliki kemampuan untuk melihat korelasi tersembunyi yang terjadi pada data yang aplikasinya digunakan pengenalan suara, pemrosesan bahasa alami, dan prediksi deret waktu. RNN sangat baik untuk masalah pemodelan urutan dengan beroperasi pada informasi input serta jejak informasi yang diperoleh sebelumnya karena koneksi berulang (Tian & dkk, 2018).

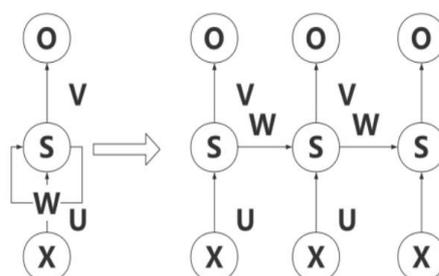
$$S_t = ((U * X_t) + (W * S_{t-1})) \quad (3.1)$$

$$O_t = (V * S_t) \quad (3.2)$$

Dengan  $S_t$  adalah memori jaringan pada waktu  $t$ ;  $U$ ,  $W$ , dan  $V$  adalah matriks bobot berbagi di setiap lapisan;  $X_t$  dan  $O_t$  mewakili input dan output pada waktu  $t$ ; dan  $f(\cdot)$  dan  $g(\cdot)$  mewakili fungsi nonlinear.

*Recurrent Neural Network* (RNN) terdiri dari unit input, unit output, dan unit tersembunyi. Model RNN pada dasarnya memiliki aliran informasi satu arah dari unit input ke unit tersembunyi, dan sintesis aliran informasi satu arah dari unit tersembunyi sementara sebelum ke unit tersembunyi waktu saat ini. Unit

tersembunyi dapat dilihat sebagai penyimpanan seluruh jaringan, yang mengingat informasi dari ujung ke ujung (Yin, 2017). Seperti yang ditunjukkan pada gambar dibawah ini:



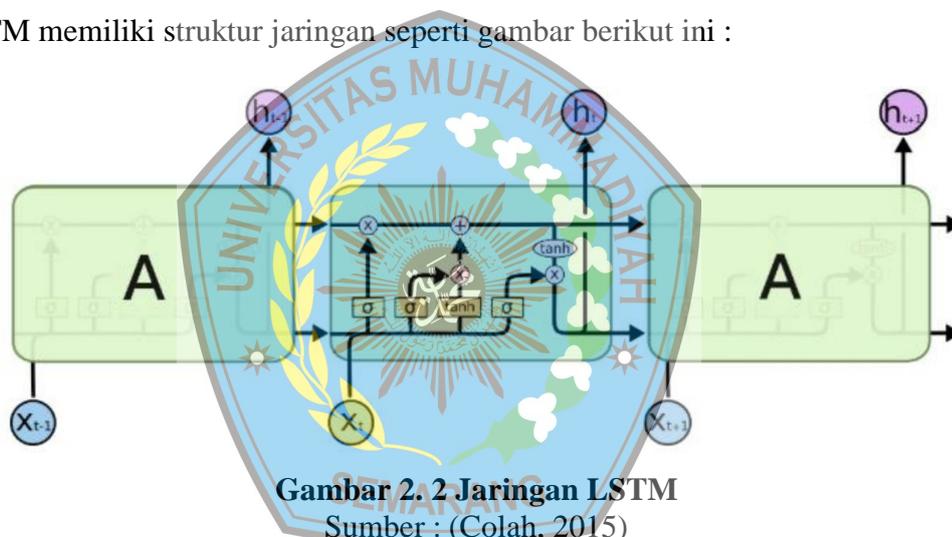
**Gambar 2. 1 Struktur RNN**

Sumber: (Tian & dkk, 2018)

Berbeda dengan koneksi bobot yang dibangun antara lapisan dalam jaringan saraf dasar, RNN dapat menggunakan status internal (memori) untuk memproses urutan input. Status tersembunyi menangkap informasi pada titik waktu sebelumnya, dan output berasal dari waktu saat ini dan memori sebelumnya. RNN berkinerja baik ketika output dekat dengan input terkait informasi dari node sebelumnya diteruskan ke node berikutnya. Secara teori, RNN juga mampu menangani ketergantungan jangka panjang. Namun, dalam aplikasi praktis, RNN tidak dapat menyimpan informasi sebelumnya dengan baik ketika interval waktu lama karena masalah gradien menghilang. Untuk mengatasi kelemahan ini dan meningkatkan kinerja RNN, sebuah tipe khusus arsitektur RNN yang disebut LSTM diusulkan oleh Hochreiter & Schmidhuber (1997) (Tian & dkk, 2018).

## 2.6. Long Short Term Memory (LSTM)

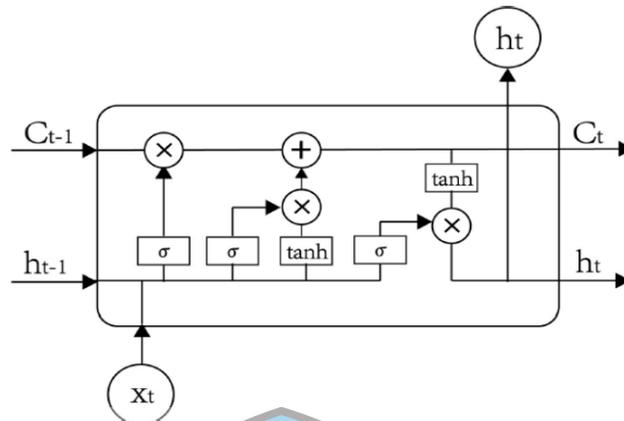
Long Short Term Memory (LSTM) menggunakan salah satu bentuk RNN yang paling umum yang dimaksudkan untuk menghindari masalah ketergantungan jangka panjang dan sesuai untuk memproses serta memprediksi deret waktu. Diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, model LSTM terdiri dari serangkaian sel memori unik yang menggantikan neuron lapisan tersembunyi dari RNN, dan kuncinya adalah keadaan sel-sel memori. LSTM memiliki struktur jaringan seperti gambar berikut ini :



**Gambar 2. 2 Jaringan LSTM**  
Sumber : (Colah, 2015)

Model LSTM menyaring informasi melalui struktur gerbang untuk mempertahankan dan memperbarui keadaan sel memori. Struktur pintunya mencakup gerbang input, *forget gate*, dan gerbang output. Setiap sel memori memiliki tiga lapisan sigmoid dan satu lapisan tanh. Lapisan sigmoid menampilkan angka antara nol dan satu, menggambarkan berapa banyak masing-masing komponen yang harus dilewati. Nilai nol berarti "jangan biarkan apa pun lewat" sementara nilai satu berarti "biarkan semuanya lewat". Sigmoid akan

bernilai nol jika terletak pada rentang  $0 \leq \sigma \leq 0,5$  dan bernilai satu jika terletak pada rentang  $0,5 < \sigma \leq 1$  (R, 2017).



**Gambar 2.3 Struktur LSTM**

Sumber : (Qiu, Wang, & Zhou, 2020)

Fungsi sigmoid ditunjukkan pada persamaan (3.3) (Ma, 2015):

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (3.3)$$

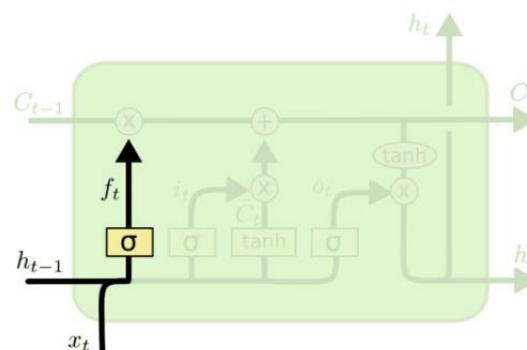
Fungsi tanh adalah sebagai berikut :

$$\tanh(x) = 2\sigma(2x) - 1 \quad (3.4)$$

Dimana :

$\sigma$  = Fungsi aktivasi sigmoid

$x$  = Data input



**Gambar 2.4 Struktur LSTM untuk Forget Gate**

Sumber: (Colah, 2015)

*Forget gate* dalam unit LSTM menentukan informasi status sel mana yang dibuang dari model. Seperti yang ditunjukkan pada Gambar 2.4, sel memori menerima output  $c$  dari momen sebelumnya dan informasi eksternal  $X_t$  dari momen saat ini sebagai input dan menggabungkannya dalam vektor panjang  $[h_{t-1}, X_t]$ . Notasi  $[h_{t-1}, X_t]$  merupakan operasi konkatenasi, artinya menambahkan baris dari  $X_t$  dengan baris dari  $h_{t-1}$  melalui transformasi  $\sigma$  menjadi:

$$f_t = \sigma (W \cdot [h_{t-1}, X_t] + X_f) \quad (3.5)$$

Keterangan:

$f$  : *Forget gate*

$\sigma$  : Fungsi sigmoid

$W$  : Nilai *weight* untuk *forget gate*

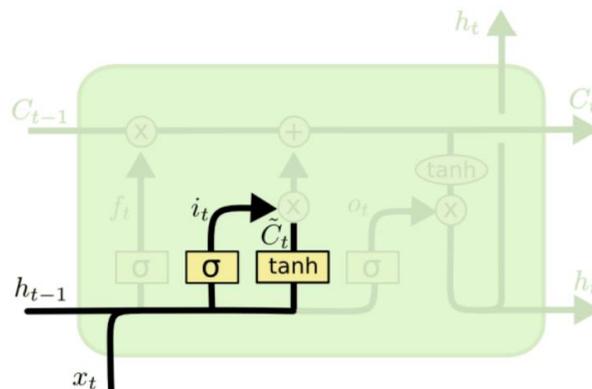
$h_{t-1}$  : Nilai output sebelum orde ke- $t$

$x$  : Nilai input pada orde ke- $t$

$b$  : Nilai bias pada *forget gate*



Fungsi utama dari *forget gate* adalah untuk merekam seberapa banyak status sel  $C_{t-1}$  dari waktu sebelumnya dicadangkan ke status sel  $C_t$  dari waktu saat ini. Gerbang akan menampilkan nilai antara 0 dan 1 berdasarkan pada  $h_{t-1}$  dan  $x_t$ , dimana 1 menunjukkan reservasi lengkap dan 0 menunjukkan pembuangan lengkap.



**Gambar 2. 5 Struktur LSTM untuk *Input Gate***

Sumber: (Colah, 2015)

*Input gate* atau gerbang input menentukan berapa banyak input jaringan waktu saat ini  $x_t$  dicadangkan ke dalam status sel, yang mencegah konten tidak signifikan dari memasuki sel memori. Gerbang input memiliki dua fungsi, pertama adalah menemukan keadaan sel yang harus diperbarui; nilai yang akan diperbarui dipilih oleh lapisan sigmoid, seperti dalam Persamaan (3.6):

$$i_t = \sigma (W \cdot [h_{t-1}, x_t ] + b_i) \quad (3.6)$$

Keterangan:

$i$  : *Input gate*

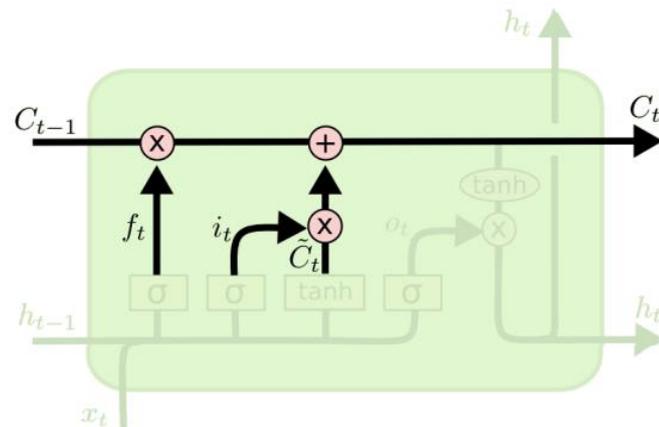
$\sigma$  : Fungsi sigmoid

$W$  : Nilai *weight* untuk *input gate*

$h_{t-1}$  : Nilai output sebelum orde ke-t

$x$  : Nilai input pada orde ke-t

$b$  : Nilai bias pada *input gate*



**Gambar 2. 6 Struktur LSTM untuk *Cell State***

Sumber: (Olah, 2015)

Sedangkan fungsi lainnya adalah untuk memperbarui informasi ke  $b$  diperbarui ke keadaan sel. Vektor kandidat baru  $C_t$  dibuat melalui lapisan tanh untuk mengontrol berapa banyak informasi baru ditambahkan, seperti dalam Persamaan (3.7). Dan Persamaan (3.8) digunakan untuk memperbarui keadaan sel dari sel memori

$$C_t = \tanh(W \cdot [h_{t-1}, x_t] + b_c) \quad (3.7)$$

Keterangan:

$C_t$  : Nilai baru yang ditambahkan ke *cell state*

*Tanh* : Fungsi *tanh*

$W$  : Nilai *weight* untuk *cell state*

$h_{t-1}$  : Nilai output sebelum orde ke- $t$

$x$  : Nilai input pada orde ke- $t$

$b$  : Nilai bias pada *cell state*

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (3.8)$$

Keterangan:

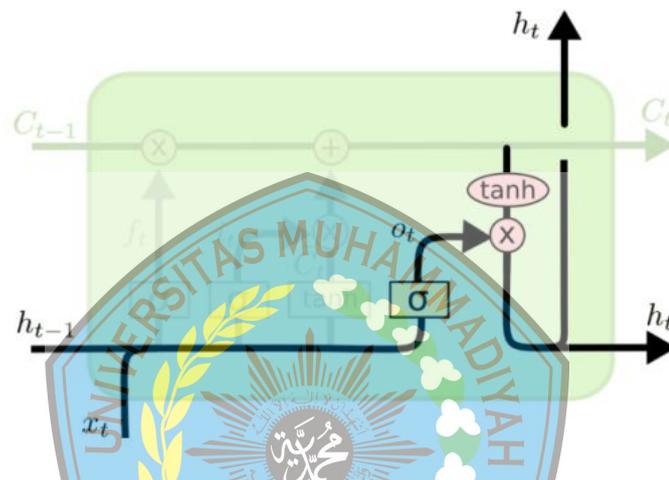
$C$  : Nilai baru yang ditambahkan ke *cell state*

$f$  : *Forget gate*

$C_{t-1}$  : *Cell state* sebelum orde ke-t

$i$  : *Input gate*

$C_t$  : Nilai baru yang ditambahkan ke *cell state*



Gambar 2. 7 Struktur LSTM untuk *Output Gate*

Sumber: (Olah, 2015)

Gerbang output mengontrol seberapa banyak keadaan sel saat ini dibuang. Informasi keluaran pertama-tama ditentukan oleh lapisan sigmoid, dan kemudian keadaan sel diproses oleh tanh dan dikalikan dengan keluaran lapisan sigmoid untuk mendapatkan bagian keluaran akhir:

$$O_t = \sigma (W_o \cdot [h_{t-1}, x] + b_o) \quad (3.9)$$

Keterangan:

$O$  : *Output gate*

$\sigma$  : Fungsi sigmoid

$W$  : Nilai *weight* untuk *output gate*

$h_{t-1}$  : Nilai output sebelum orde ke-t

$x$  : Nilai input pada orde ke-t

$b$  : Nilai bias pada *output gate*

Nilai output akhir sel didefinisikan sebagai:

$$h_t = O_t * \tanh(C_t) \quad (3.10)$$

Keterangan:

$h$  : Nilai output orde ke t

$O$  : *Output gate*

$\tanh$  : Fungsi *tanh*

$C$  : *Cell state*

Algoritma yang digunakan dalam LSTM adalah sebagai berikut (Brownlee, 2017):

- 1) Menghitung nilai sigmoid dan tanh.
- 2) Mengubah data yang akan digunakan menjadi *supervised learning problem*. *Supervised learning problem* adalah algoritma yang digunakan untuk mempelajari fungsi pemetaan dari input ke output. Karena data yang digunakan adalah data *time series* maka inputnya adalah data hari kemarin (t-1) dan outputnya adalah data hari ini (t).
- 3) Melakukan normalisasi menggunakan *min-max scaling*.
- 4) Menghapus variabel yang tidak dibutuhkan.
- 5) Membagi data menjadi data *training* dan *testing*.
- 6) Membuat model LSTM yang akan digunakan dengan menentukan banyaknya *neuron* dan *epoch* yang akan digunakan. Dalam model LSTM akan dilakukan proses berupa:

- a. Menghitung nilai *forget gate*.
  - b. Menghitung nilai *input gate*.
  - c. Memperbarui memori yang berada pada *cell*.
  - d. Menghitung *output gate* dan nilai output akhir.
- 7) Melakukan evaluasi terhadap model LSTM yang telah dibuat.

## 2.7. Genetic Algorithm (GA)

Algoritma Genetika merupakan suatu teknik optimasi yang didasarkan pada prinsip genetik dan seleksi alam. Dalam Algoritma Genetika populasi terbentuk dari banyak individu yang berkembang sesuai aturan seleksi spesifik dengan memaksimalkan *fitness* (Haupt dan Haupt, 2004). Algoritma ini juga digunakan untuk mendapatkan nilai global optimum dengan cara melakukan perulangan atau iterasi pada konsep evolusi darwin.

Menurut Trevino dan Falciani (2006) terdapat 7 tahapan untuk menjalankan algoritma genetika, yaitu :

1. Membentuk populasi awal terdiri dari beberapa kromosom yang didalamnya memuat gen kromosom pada algoritma genetika yang digunakan untuk menunjukkan kandidat sekelompok gen yang dapat digunakan sebagai solusi permasalahan. Gen pada algoritma genetika berisi variabel yang ingin dioptimumkan.
2. Masing-masing kromosom dalam populasi dievaluasi kemampuannya dengan menggunakan fungsi *fitness*. Pada penelitian ini fungsi fitnessnya berupa kesalahan klasifikasi.

3. Ketika sebuah kromosom memiliki nilai fitness lebih optimum daripada nilai inisialnya, maka kromosom dihentikan, namun apabila tidak maka tahapan analisis dilanjutkan ke tahap 4. Nilai fitness terkecil yang dipilih sebagai solusi permasalahan dari penelitian ini, karena fungsi fitness yang digunakan adalah tingkat kesalahan klasifikasi.
4. Memilih kromosom dengan nilai fitness yang optimum yang dijadikan orang tua.
5. Mengkombinasikan informasi genetika yang ada dalam replikasi orang tua melalui pindah silang. Dua induk secara random dipilih dan digunakan untuk membentuk dua kromosom baru.
6. Melakukan mutasi untuk memperkenalkan unsur gen baru pada kromosom secara acak.
7. Tahapan diulangi dari tahapan 2 sampai kromosom yang memberikan nilai fitness paling optimum atau sudah mencapai konvergen.

### **2.7.1 Pengkodean**

Pengkodean adalah proses menggambarkan bentuk gen dalam kromosom. Pengkodean dapat berupa bits, number, trees, arrays, list, dan lain-lain. Pengkodean yang dilakukan pada penelitian ini adalah pengkodean bits (biner) untuk seleksi variabel.

### **2.7.2 Fitness**

Fungsi fitness digunakan untuk mengukur tingkat kebaikan atau kesesuaian suatu solusi yang dicari. Fungsi fitness bisa berhubungan dengan

langsung fungsi tujuan, atau bisa juga sedikit modifikasi terhadap fungsi tujuan. Sejumlah solusi yang dibangkitkan dalam populasi akan dievaluasi menggunakan fungsi fitness. Fungsi fitness ( $F(x)$ ) yang digunakan adalah:

$$F(X) = \frac{1}{1+f(x)} \quad (3.11)$$

dimana  $f(x)$  adalah fungsi tujuan dari masalah yang dapat terselesaikan. Untuk kasus minimasi, jika didapatkan ( $x$ ) yang kecil maka nilai fitnessnya besar. Sebaliknya, untuk kasus maksimasi, fungsi fitness-nya bias menggunakan nilai ( $x$ ) sendiri, jadi  $F(x) = f(x)$  (Santosa dan Willy, 2011).

### 2.7.3 Seleksi Orang Tua

Desiani dan Arhami (2006) mengemukakan bahwa seleksi orang tua bertujuan untuk memberikan kesempatan reproduksi bagi anggota populasi yang memiliki nilai fitness tinggi. Pemilihan dua buah kromosom dalam suatu populasi sebagai orang tua yang akan dipindahsilangkan biasanya secara proporsional sesuai dengan nilai fitness masing-masing. Metode umum yang dipakai adalah Roulette Wheel (Roda Roulette). Pada metode ini, masing-masing kromosom menempati potongan lingkaran pada Roda Roulette secara proporsional sesuai dengan nilai fitnessnya (Suyanto, 2005). Sebuah kromosom yang nantinya akan terpilih adalah apabila bilangan random yang dibangkitkan berada dalam nilai interval kumulatifnya. Nilai kumulatif ini didapatkan dari membagi nilai fitness dari tiap kromosom dengan total nilai fitness keseluruhan.

### 2.7.4 Pindah Silang

Pindah silang merupakan metode pengkombinasian 2 orang tua kromosom untuk membentuk generasi baru. Orang tua diperoleh dari tahapan seleksi orang tua menggunakan RWS. Kromosom dengan nilai fitness yang tinggi akan dijadikan sebagai orang tua. Terdapat beberapa metode untuk pindah silang yaitu, single point, two points, arithmetic, intermediate, dan scattered. Penelitian yang dilakukan oleh Bocko, Nohajova, & Harcarik (2011) terkait perbandingan hasil beberapa metode pindah silang dengan berbagai macam proses seleksi orang tua memberikan hasil bahwa untuk seluruh proses seleksi orang tua, metode pindah silang scattered memberikan performa paling bagus. Oleh karena itu, pada penelitian ini digunakan metode scattered untuk proses pindah silang. Metode ini menghasilkan kromosom baru yang berisi random nilai 0 dan 1, nilai 1 menunjukkan gen yang berasal dari orang tua pertama dan nilai 0 menunjukkan gen yang berasal dari orang tua kedua (Kusumawardani, 2015).

### **2.7.5 Mutasi**

Metode yang digunakan untuk mutasi pada penelitian ini adalah metode uniform. Tujuan mutasi adalah untuk mendapatkan keberagaman gen. Terdapat 2 tahapan dalam metode ini, pertama adalah memilih sebagian gen dari kromosom yang akan dimutasi dengan peluang sebesar 0.1, kedua mengganti gen yang terpilih dengan bilangan random yang dibangkitkan dari batas bawah dan batas atas nilai gen dalam kromosom.

### **2.7.6 Elitisme**

Suatu individu yang memiliki nilai fitness tertinggi tidak akan selalu terpilih karena proses seleksi dilakukan secara random. Oleh karena itu perlu

dilakukan elitisme, yaitu suatu prosedur pengopian individu agar individu yang bernilai fitness tertinggi tidak hilang selama proses evolusi (Suyanto, 2005).

## 2.8. *Genetic Algorithm-Long Short Term Memory (GA-LSTM)*

Pendekatan *hybrid* dari jaringan LSTM yang mengintegrasikan GA untuk menemukan jendela waktu yang disesuaikan dan jumlah unit LSTM untuk prediksi deret waktu keuangan. Sejak Jaringan LSTM menggunakan informasi masa lalu selama proses pembelajaran, jendela waktu yang dipilih dengan tepat memainkan peran penting dalam kinerja yang menjanjikan. Jika jendela terlalu kecil, model akan melakukannya mengabaikan informasi penting, sementara, jika jendelanya terlalu besar, model akan dipasang berlebihan pada data pelatihan (Chung, 2020).

Penelitian ini terdiri dari dua tahap, yaitu sebagai berikut. Tahap pertama percobaan melibatkan merancang parameter jaringan yang sesuai untuk jaringan LSTM. Menggunakan jaringan LSTM dengan lapisan masukan berurutan diikuti oleh dua lapisan tersembunyi, dan jumlah optimal neuron tersembunyi di masing-masing lapisan tersembunyi diselidiki oleh GA. Dalam model LSTM-RNN, fungsi tangen hiperbolik digunakan sebagai fungsi aktivasi dari node input dan node tersembunyi. Fungsi tangen hiperbolik adalah a diskalakan fungsi sigmoid, dan mengembalikan nilai masukan ke dalam rentang antara  $-1$  dan  $1$ . Aktivasi Fungsi node keluaran ditetapkan sebagai fungsi linier, karena tujuan kita adalah prediksi penutupan harga hari berikutnya yang dapat dirumuskan sebagai masalah regresi. Bobot awal jaringan ditetapkan sebagai nilai acak, dan bobot

jaringan disesuaikan dengan menggunakan "Adam" berbasis gradien pengoptimal, yang terkenal dengan kesederhanaan, keterusterangan, dan efisiensi komputasi. Metode ini cocok untuk masalah yang memiliki data dan parameter besar, dan juga memiliki kekuatan dalam menangani masalah non-stasioner dengan gradien yang sangat bising dan jarang.

Algoritma pencarian evolusioner, GA, untuk diselidiki ukuran jendela waktu yang optimal dan faktor arsitektur jaringan LSTM. Pada tahap kedua, berbagai ukuran jendela waktu dan jumlah unit LSTM yang berbeda dari setiap lapisan tersembunyi diterapkan untuk mengevaluasi kesesuaian GA. Populasi yang tersusun dengan kemungkinan solusi adalah diinisialisasi dengan nilai acak, sebelum operator genetik mulai menjelajahi ruang pencarian. Itu kromosom yang digunakan dalam penelitian ini dikodekan dalam bit biner yang mewakili ukuran jendela waktu dan jumlah sel LSTM. Berdasarkan populasi, seleksi dan rekombinasi operator dimulai untuk mencari solusi terbaik. Solusi dievaluasi oleh fungsi *fitness* yang telah ditetapkan sebelumnya, dan string dengan performa menonjol dipilih untuk reproduksi. Fungsi *fitness* sangat penting bagian dari GA, dan harus dipilih dengan hati-hati.

## **2.9. Adaptive Momen Estimation (ADAM)**

Optimasi berbasis gradien stokastik adalah kepentingan praktis inti dalam banyak bidang sains dan teknik. Banyak masalah dalam bidang ini dapat dilemparkan sebagai optimalisasi beberapa fungsi tujuan berstandarisasi skalar yang membutuhkan maksimalisasi atau minimalisasi berkenaan dengan

parameternya. Adam merupakan algoritma optimasi yang dikembangkan dengan memanfaatkan kelebihan dari algoritma *Adaptive Gradient* (AdaGrad) yang bekerja dengan baik dengan gradien jarang dan *Root Mean Square Propagation* (RMSProp) yang bekerja dengan baik secara *online* dan non pengaturan stasioner. Adam tidak hanya mengadaptasi tingkat pembelajaran parameter berdasarkan rata-rata pertama (*mean*) seperti dalam RMSProp, namun Adam juga menggunakan rata-rata kedua dari gradien (*varians uncentered*). Algoritma menghitung rata-rata pergerakan eksponensial dari gradien dan gradien kuadratnya, dan parameter  $\beta_1$  dan  $\beta_2$  mengontrol tingkat peluruhan rata-rata pergerakan (Aldi, Jondri, & Aditsania, 2018).

Beberapa keuntungan Adam adalah bahwa besarnya pembaruan parameter tidak sama dengan penskalaan gradien, ukurannya kira-kira dibatasi oleh *hyperparameter stepsize*, tidak memerlukan alat tulis stasioner. *Adaptive Moment Estimation* (Adam) adalah metode yang menghitung *adaptive learning rate* untuk setiap parameter. Nilai parameter yang direkomendasikan adalah  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , dan  $\epsilon = 10^{-8}$  dengan  $\beta_1 = \beta_2 =$  tingkat penurunan eksponensial dan  $\epsilon =$  nilai epsilon untuk update parameter (Kingma & Ba, 2015)

### **2.10. Root Mean Square Error (RMSE)**

Sebuah model peramalan yang dibangun perlu dilakukan evaluasi untuk mengukur tingkat akurasi atau tingkat kesalahan hasil prakiraan yang dibuat. Menurut (Makridakis dkk, 1982) salah satu metode alternatif untuk mengevaluasi ukuran kesalahan dalam peramalan adalah nilai tengah akar kuadrat atau Root Mean Square Error (RMSE). Evaluasi RMSE dihitung berdasarkan nilai rata-rata

dari jumlah kuadrat kesalahan antara hasil prakiraan dengan nilai aktual. Nilai digunakan untuk mengukur besarnya kesalahan yang dihasilkan oleh model prakiraan. Suatu model prakiraan mendekati variasi nilai aktualnya jika hasil perhitungan RMSE rendah. Persamaan yang digunakan untuk menghitung parameter tersebut adalah sebagai berikut :

$$RMSE = \sqrt{\frac{1}{m \times n} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [f(i,j) - g(i,j)]^2} \quad (3.12)$$

### 2.11. Mean Absolute Percentage Error (MAPE)

*Mean Absolute Percentage Error* (MAPE) adalah nilai absolute dari persentase error data terhadap mean. Persamaannya ditulis seperti di bawah ini:

$$MAPE = \frac{\sum_{i=1}^n \frac{|\hat{Y}_i - Y_i|}{Y_i}}{n} * 100\% \quad (3.13)$$

Keterangan :

n = Jumlah data

$\hat{Y}_i$  = Nilai prediksi pada periode ke-i

$Y_i$  = Nilai data aktual pada periode ke-i

Menurut Chang, Wang dan Liu (2007) kriteria dari nilai MAPE adalah sebagai berikut:

**Tabel 2. 1 Kriteria MAPE**

MAPE	Pengertian
<10%	Kemampuan Peramalan Sangat Baik
10%-20%	Kemampuan Peramalan Baik
20%-50%	Kemampuan Peramalan Cukup

>50%

Kemampuan Peramalan Buruk

---

