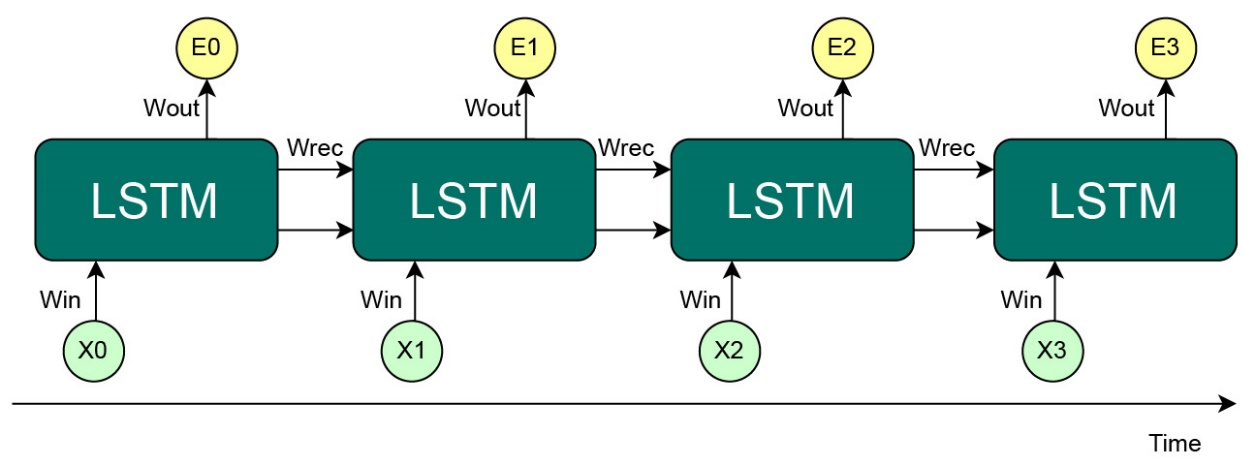
# BAB III METODE USULAN

## 3.1 Desain Rancangan

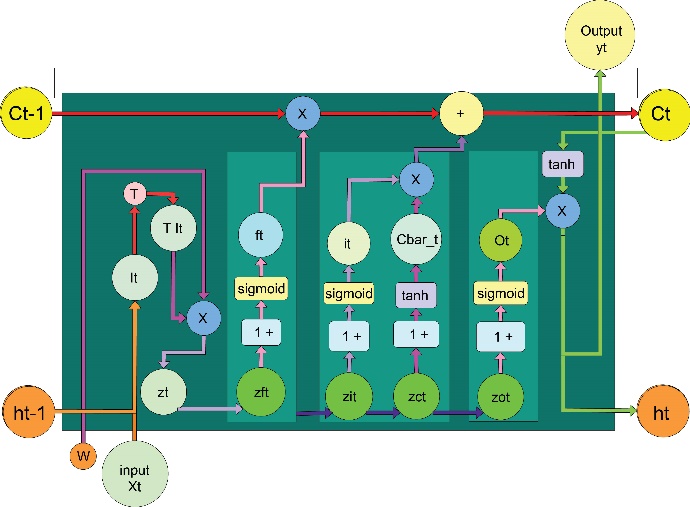
Desain rancangan sistem dibuat agar sistem yang dibangun akan lebih mudah untuk meganalisa jika terjadi kesalahan ataupun ketidaksesuaian dengan kondisi yang diharapkan.

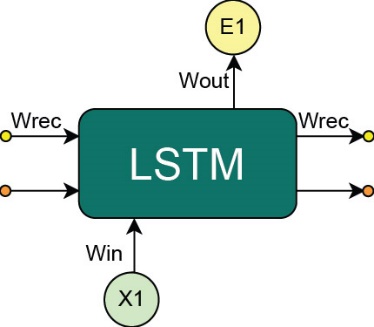
### 3.1.1 Rancangan Arsitektur LSTM

Arsitektur ini adalah Rancangan LSTM yang mendeskripsikan alur algoritma LSTM dari proses *forward* sampai *backward*, Alur algorimanya adalah sebagai berikut :



Gambar 3.11 Arsitektur LSTM forward

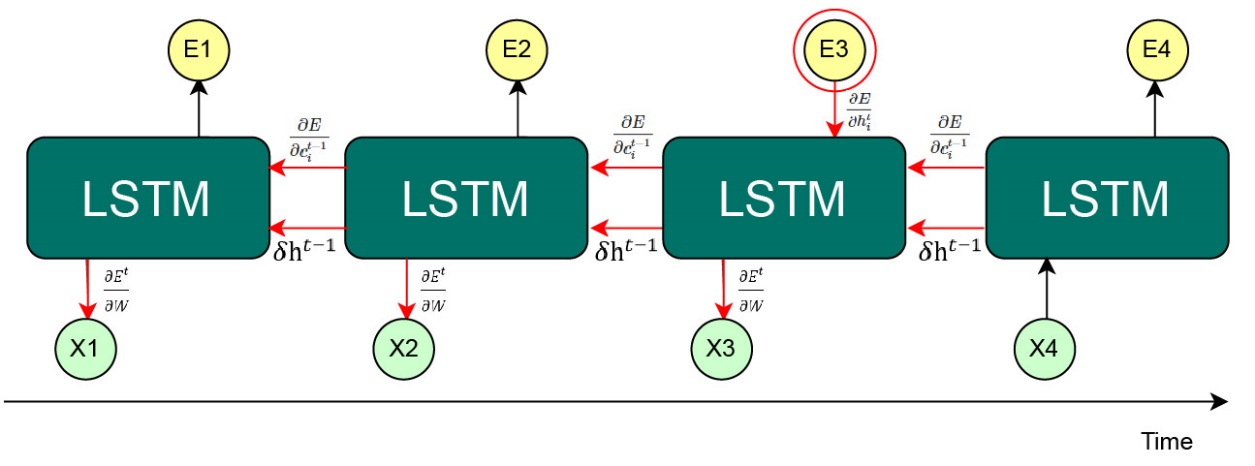


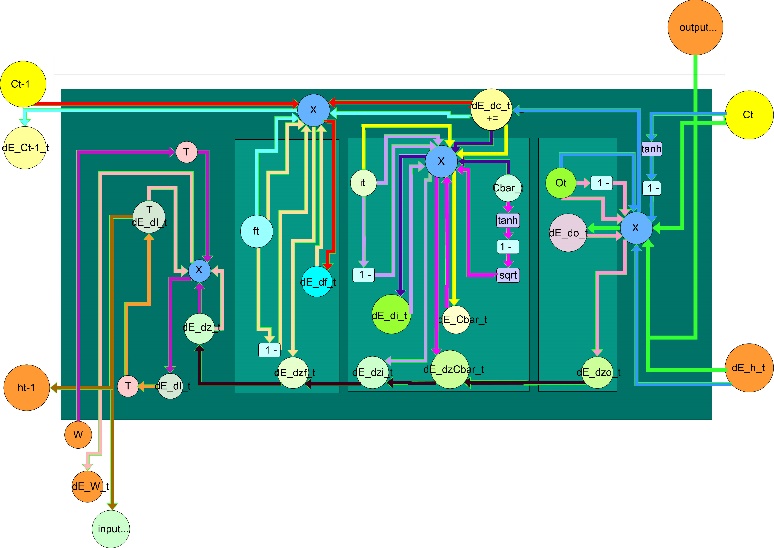


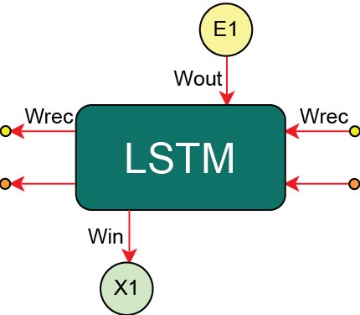
Gambar 3.12 Penjabaran dari Hidden Layer LSTM

Pada gambar 5 setiap urutan terdapat hidden layer yang berhubungan satu sama lain dan terdapat gerbang – gerbang yang bertujuan untuk menyeleksi informasi yang tidak perlu diingat dalam urutan selanjutnya seperti ditunjukan pada gambar 3.2.

Dari tahap forward yang ditunjukan pada gambar 3.1 didapatlah bobot baru (, bobot tersebut memiliki hasil *error* yang besar, jadi dibutuhkanlah proses *Backward* untuk mendapatkan *error* bobot terbaik.



 Gambar 3.13 Arsitektur LSTM Backward



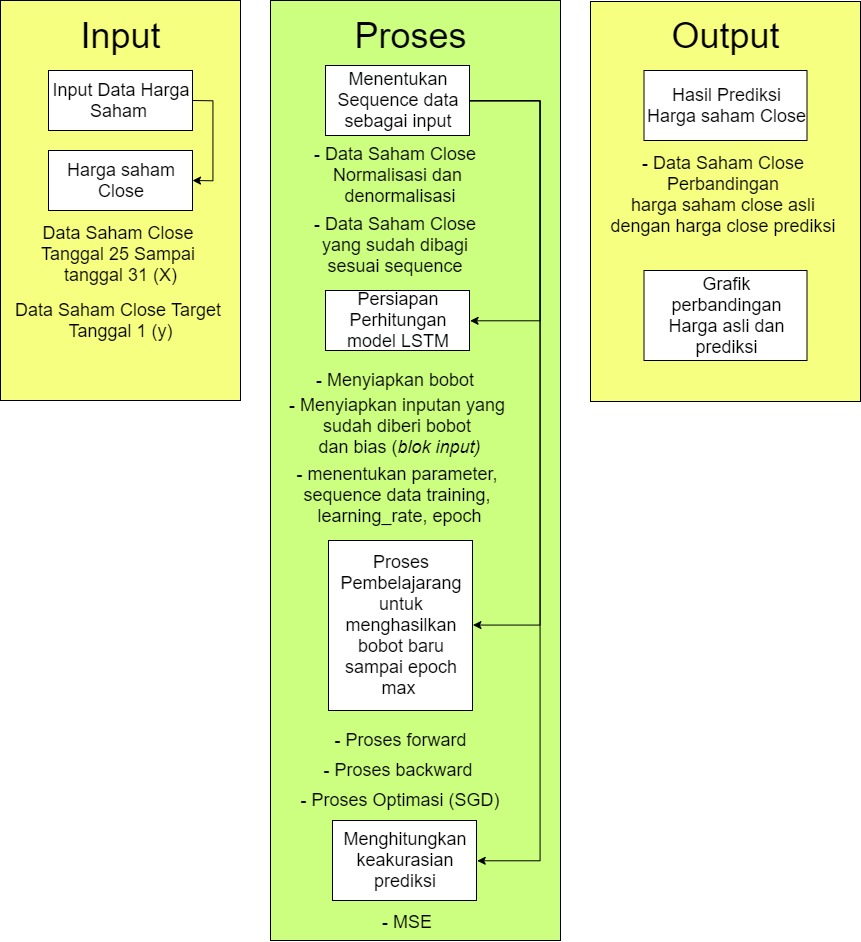
Gambar 3.14 Penjabaran dari Hidden Layer LSTM Backward

Pada gambar 3.3 ditunjukan langkah pertama di mulai dari *output* terakhir berproses menuju ke urutan *hidden* *layer* sampai urutan pertama*,* didalam hidden layer memberikan 3 keluaran yaitu, *derivative block input* , *derivative* keadaan penghubung *(cell state) ,* dan bobot *derivative* .

*Derivative block input* dan *derivative* keadaan penghubung *(cell state) ,* ini akan menghubungkan dari urutan ke urutan sebelumnya sampai ke urutan pertama untuk mendapatkan *derivative minimum,* Dan tiap urutan hidden layer menghasilkan bobot *derivative*  yang akan di proses lagi dengan perhitungan update bobot dengan metode *Gradient descent weight.*

### 3.1.2 Diagram I-P-O

Rancangan sistem yang mendeskripsikan alur sistem dari awal hingga akhir dapat dilihat pada Gambar 3.5 :

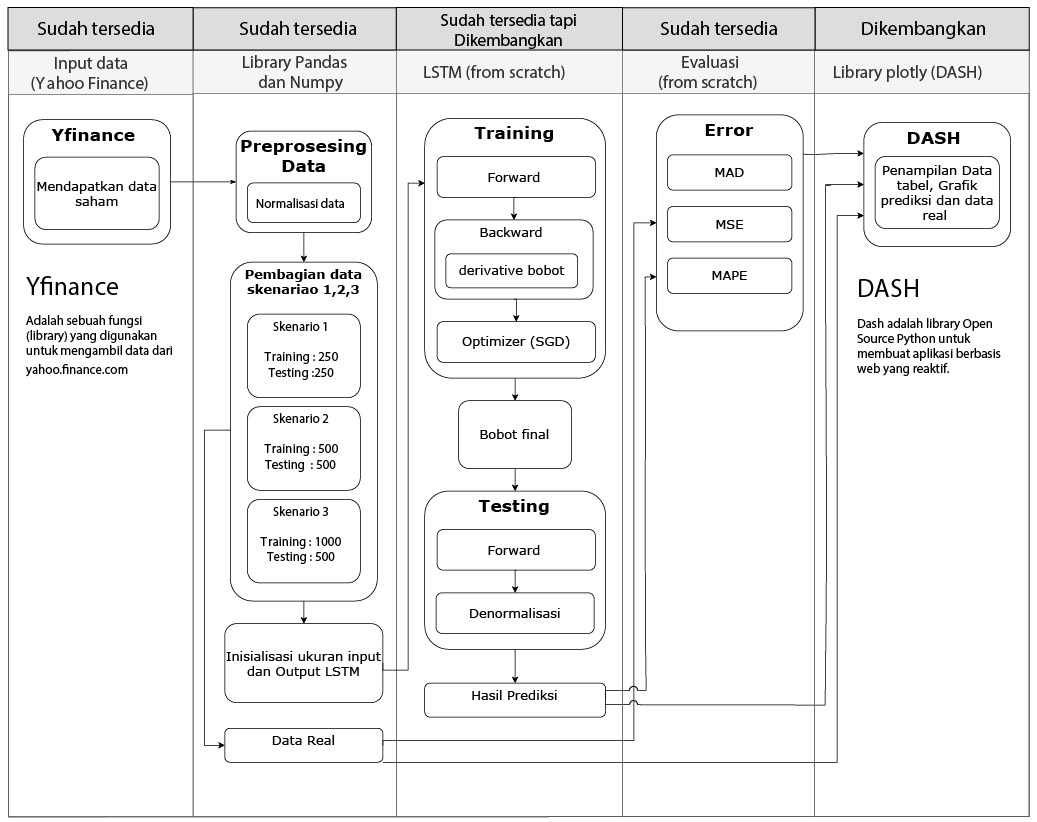


Gambar 3.15 Diagram IPO Arsitektur Sistem

Rancangan sistem menjelaskan proses berjalannya program. Dimulai dengan pengambilan data harga saham. Kemudian diolah menjadi dataset untuk diproses menjadi data *training*.

Tahap proses menginisialisasi data deret waktu *(sequence)*, Tahap selanjutnya mempersiapkan perhitungan LSTM dengan menyiapkan bobot, menyiapkan masukan yang sudah diberi bobot dan bias *(block input)* didalam proses training ada dua pembelajaran yaitu *forward propagation* dan *backward propagation* dari proses *backpropagation* bertujuan untuk mendapatkan nilai *error* yang terbaik. Hasil *training* berbentuk bobot baru yang akan diguakan untuk pemodelan, setelah mendapatkan model proses prediksi harga didapatkan, dilakukan evaluasi dengan metode evaluasi *mse* dengan membandingkan masing-masing hasil prediksi dengan data aktual. Evaluasi ini digunakan untuk mengukur seberapa akurat hasil prediksi dengan menggunkan LSTM. Output proses menampilkan hasil prediksi dan grafik perbandingan harga asli dan harga prediksi.

### 3.1.3 Diagram Pengembangan.

Rancangan sistem yang mendeskripsikan alur system dalam pemrograman dari awal hingga akhir

Gambar *3.* Diagram Pengembangan

1. Input data menggunakan library yfinance yang bertujuan untuk mengambil data dari Yahoofinance dan mengembalikan data dalam bentuk excel maupun pandas\_datareader().
2. *Library* Pandas dan Numpy
3. *Library* Pandas adalah *library open source*, struktur data yang mudah digunakan dan alat analisis data untuk bahasa pemrograman Python.
4. *Library* Numpyadalah *library open source* untuk komputasi ilmiah dengan python yang mudah digunakan salah satunya menghitung objek array N-dimensi yang baik.

Dalam menyiampkan data untuk proses LSTM digunakan library dimana library pandas digunakan untuk membaca data dan menyeleksi data yang akan diekseskusi dalam program. Dan libarary Numpy digunakan untuk proses matematis seperti oprasi matriks, penggabungan matriks, dan lainnya.

1. Metode LSTM (from scratch)

Didalam system ini menggunakan code yang sudah dikembangkan dari *library KERAS* pada modul LSTM menjadi modul dari awal proses metode LSTM berkerja sampai akhir. Jadi didalam code ini terdapat proses forward, backward, dan optimizer yang bisa kita lihat hasil dari proses yang berjalan didalam proses LSTM tersebut.

Dari code tersebut penelitian ini mengembangkan code ini untuk bisa menampilkan proses hasilnya tiap langkah dari proses LSTM dalam bentuk Excel dengan bantuan library pandas.

1. Evaluasi (from scratch)

Didalam system ini menggunakan evaluasi MAD, MSE, MAPE yang ditulis dengan code dari awal sesuai rumus yang tertera dalam buku.

1. GUI (Graphical user interface)

System ini memberikan GUI yang interaktif dengan menggunakan library Dash yang sering digunakan dalam menampilakan hasil penelitihan yang berbasis web.

## Analisa Kebutuhan

### 3.2.1 Kebutuhan *Hardware*

Hardware adalah komponen pada komputer yang terlihat dan dapat disentuh yang digunakan untuk melakukan pembuatan sistem dan penginputan data. Hardware yang digunakan dalam pembuatan sistem penerapan metode *LSTM* untuk penerapan metode *LSTM* dalam prediksi harga saham adalah sebagai berikut :

Laptop dengan spesifikasi sebagai berikut :

* Processor : Intel® Core™ i5-4200U CPU @ 1.60GHz
* Memori : 8 RAM

### 3.2.2 Kebutuhan *Software*

Software adalah komponen yang penting dalam komputer. software memiliki tugas untuk menjalankan perintah. Software ini dibuat dengan Bahasa pemograman oleh programmer yang kemudian di hubungkan dengan hardware. Software yang yang digunakan dalam pembuatan sistem penerapan metode *LSTM* untuk penerapan metode *LSTM* dalam prediksi harga saham adalah sebagai berikut :

1. Operating sistem : Windows 10 Pro 64-bit
2. *Program Tool* : IDLE Python
3. *Framework Python* : Dash, Pandas, Yfinance, Numpy.
4. *Text Editor* : Visual studio code

## 3.3 Analisa Input

Inptan dalam penelitihan ini menggunakan data yang terdapat di Alphavantag dan Yahoo finance yaitu data harga saham PT Semen Indonesia (Persero) Tbk, data yang digunakan adalah data harga saham, Jumlah data yang terhimpun yaitu sebanyak 1500 data.

## 3.4 Analisa Proses

Penelitihan dalam menerapakan metode LSTM terdapat beberapa proses yang diataranya yaitu :

1. Pembuatan data untuk LSTM:
   * Normalisasi data
   * data deret waktu yang dinormalisasi
2. Pengembangan Model:
   * LSTM adalah jenis Jaringan Syaraf Berulang (RNN).

Inisialisasi panjang urutan (*sequence)* :

*Forward Propagation*

*Backpropagation* kesalahan

Pembelajaran jaringan

c. Prediksi harga saham.

## 3.5 Analisa Output

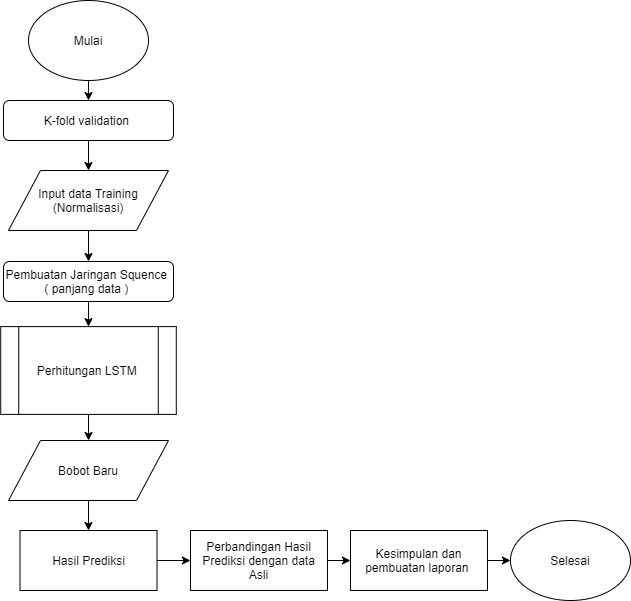
Hasil dari prediksi harga saham, saham akan ditampilkan dalam bentuk grafik perbandingan harga dari harga prediksi dengan harga aktual dalam meprediksi harga saham.

## 3.6 Desain Rancangan Sistem

Perancangan sistem yang dibuat untuk membangun sistem prediksi dengan *flowchart* yang akan diimplementasikan. Tujuan agar dalam pembuatan sistem akan lebih terkonsep dan memiliki acuan sehingga akan memudahkan ketika melakukan implemtasi pada bahasa pemrograman.

### 3.6.1 Flowchart Diagram

*Flowchart* merupakan kerangka untuk menunjukkan alur sistem.Diagram ini dapat memberi solusi langkah demi langkah untuk menyelesaikan masalah yang ada dalam algoritma tersebut.Berikut merupakakn *flowchart* sistem secara umum.



Gambar 3.17 Flowchart Sistem Keseluruhan

Keterangan dari Gambar 3.7 Secara garis besar, peramalan harga saham menggunakan Metode *LSTM* dapat dituliskan sebagai berikut :

1. Mulai
2. Masukan Inputan

Variabel input yang digunakan dalam penelitian ini adalah data jumlah harga saham dari tahun 2014 sampai 2020, yaitu data *close*.

1. Menentukan panjang urutan (*Squence length)*

Karena dalam pasar modal dalam seminggu ada 5 hari kerja, maka data yang digunakan untuk memprediksi adalah 5 hari kedepan. Jadi perlu dilakukan penentuan panjang urutan dalam pembagian data tiap 5 hari sebagai masukan dan 1 output.

1. Normalisasi Data

Proses Normalisasi yang digunakan dalam sistem ini menggunakan normalisasi minimum-maximum. Data-data yang ada dilakukan normalisasi dengan membagi nilai data tersebut dengan nilai *range* data (nilai data maksimum-nilai data minimum). Normalisasi data *input* bertujuan untuk menyesuaikan nilai *range* data dengan fungsi aktivasi dalam sistem *LSTM*. Ini berarti nilai kuadrat *input* harus berada pada *range* 0 sampai 1. Sehingga *range input* yang memenuhi syarat adalah nilai data *input* dari 0 sampai 1 atau dari –1 sampai 1. Oleh karena itu *output* yang dihasilkan pun akan berada pada *range* 0 sampai 1. kemudian untuk mendapatkan nilai sebenarnya dari *output* perlu dilakukan proses denormalisasi.

1. *Multiple Train-Test Split (K - Fold Validation)*

Proses membagi data menjadi data *trining* dan data *testing.* Yang di bagi menjadi 3 skenario.

1. Perhitungan *LSTM*

Terdiri dari beberapa tahap yaitu fase pertama propagasi maju. Kedua propagasi mundur dan yang ketiga perubahan bobot. Ketiga fase tersebut diulang terus hingga kondisi yang diinginkan terpenuhi (jumlah iterasi atau kesalahan *error*). Dalam menggunakan algoritma LSTM, proses pelatihan dengan proses pengujian memiliki langkah yang berbeda. Dalam proses pelatihan terdiri dari dua proses utama, *feed forward* dan *backpropagation of error* atau *backward*. Sedangkan untuk proses pengujian hanya menggunakan proses *feed forward*.

1. Menghasilkan *output*

*Output* yang dihasilkan pada penelitian ini yaitu harga saham periode berikutnya yang sudah dalam proses denormalisasi.

1. Selesai

### 3.6.1 Flowchart Multiple Train-Test Split

Di bawah ini merupakan *Flowchart* sistem proses *K-Folds Cross Validation.*

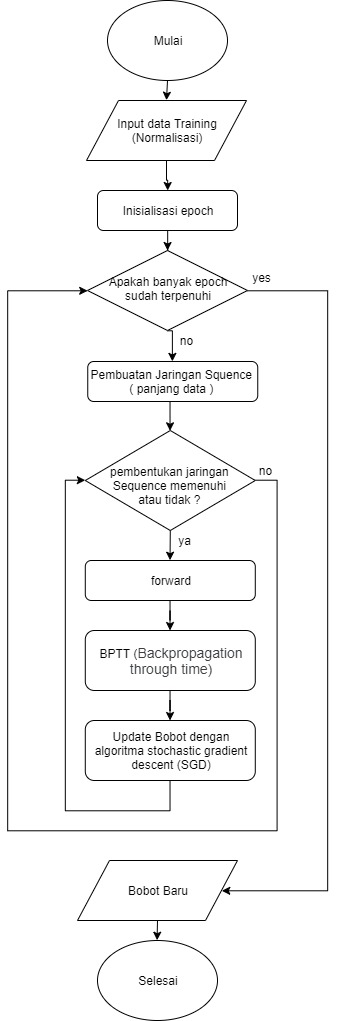
Gambar 3.18 Multiple Train-Test Split

Menentukan data *training* dan data *testing* pada penelitian ini menggunakan *Multiple Train-Test Split*.

Misal ditentukan pembagian 3 skenario sehingga akan didapat *dataset* skenario1, skenario 2, dan skenario 3. Dari 3 skenario tersebut akan dipilih 2 menjadi *data training* dan 1 menjadi *data testing.* Misalkan untuk data *training* dipilih skenario *1* dan skenario *2*, sedangkan skenario *3* menjadi data *testing*. Semakin banyak pembagian skenario semakin banyak dan semakin bervariasi uji coba yang dilakukan akan tetapi pengerjaan suatu sistem akan semakin lama.

### 3.6.2 Flowchart Tahap Pelatihan LSTM

Tahap ini adalah proses pengenalan pola-pola data yang telah dinormalisasi agar sistem dapat menentukan bobot-bobot yang dapat memetakan antara data input dengan data target output yang diinginkan.

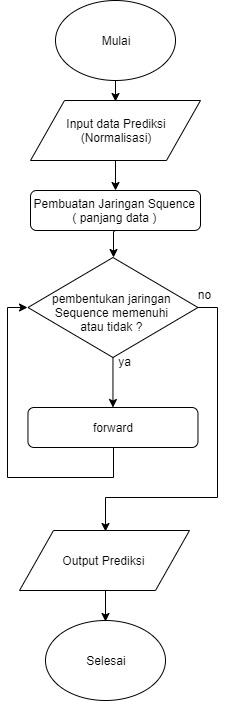


Gambar 3.19 Flowchart pelatihan LSTM

Pelatihan ini menggunakan dua looping yaitu looping pertama digunakan seberapa banyak pelatihan dalam setiap urutan (*sequence*) dan lopping kedua digunakan untuk menghitung proses pembelajaran dalam pembelajaran LSTM menerapakan BPTT *(Backpropagation through time)* dan *update* bobot menerapkan *Stochastic* *gradient descent* (SGD)

### 3.6.3 Flowchart Tahap Pengujian LSTM

Setelah bobot yang terbaik pada tahap pelatihan didapat, maka nilai pembobot tersebut digunakan untuk mengolah data masukan untuk menghasilkan keluaran yang sesuai. Hal ini digunakan untuk menguji apakah pembelajaran LSTM dapat bekerja dengan baik yaitu dapat memprediksi pola data yang telah dilatihkan dengan tingkat kesalahan yang kecil.



Gambar 3.20 Flowchart pengujian LSTM

## Rancangan Skenario Metode *LSTM*

Objek penelitian yang akan dilakukan analisis pada penelitian ini adalah peramalan harga saham pada PT Semen Indonesia (Persero) Tbk yang terdaftar di Bursa Efek Indonesia. Data yang diambil adalah data close (Harga saham penutupan yang ditetapkan pada satu hari transaksi sebelum hari transaksi yang akan diramalkan). Data yang diambil sebanyak 1500 data harga saham mulai dari tanggal (09/01/2014) sampai (08/01/2020).

Variabel input yang digunakan untuk contoh perhitungan manual dalam penelitian ini adalah data harga saham *close* dengan menentukan panjang urutan (*Squence length).* Karena dalam pasar modal dalam seminggu ada 5 hari kerja, maka data yang digunakan untuk memprediksi adalah 5 hari kedepan. Jadi perlu dilakukan penentuan panjang urutan dalam pembagian data tiap 5 hari sebagai masukan dan 1 output.

Contoh data Awal panjang urutan dalam pembagian data tiap 5 hari dengan banyak data total 10 data, 5 data untuk *training* dan 5 data untuk *testing*.

Tabel 2 Data Harga Saham PT Semen Indonesia (Persero) Tbk

|  |  |
| --- | --- |
| Date | Close |
| 20/12/2019 | 12375 |
| 23/12/2019 | 12250 |
| 26/12/2019 | 12325 |
| 27/12/2019 | 12225 |
| 30/12/2019 | 12000 |
| 02/01/2020 | 12200 |
| 03/01/2020 | 12325 |
| 06/01/2020 | 12275 |
| 07/01/2020 | 12300 |
| 08/01/2020 | 12000 |

Tabel 3 Data Training Harga close Saham

|  |  |
| --- | --- |
| Date | Close |
| 20/12/2019 | 12375 |
| 23/12/2019 | 12250 |
| 26/12/2019 | 12325 |
| 27/12/2019 | 12225 |
| 30/12/2019 | 12000 |
| 02/01/2020 | 12200 |

Tabel 4 Data Testing Harga Close Saham

|  |  |
| --- | --- |
| Date | Close |
| 02/01/2020 | 12200 |
| 03/01/2020 | 12325 |
| 06/01/2020 | 12275 |
| 07/01/2020 | 12300 |
| 08/01/2020 | 12000 |

Dari tabel *2* diberikan 5 data harga saham yang belum di *normalisasi* kan, Rumus yang digunakan dalam proses Normalisasi

Tabel 5 Data Training dan data Testing Harga close Saham dinormalisasi

|  |  |  |  |
| --- | --- | --- | --- |
| t | *Training Close* | | *Testing Close* |
| 1 | | 0,0303030 | 0,0161616 |
| 2 | | 0,0202020 | 0,0262626 |
| 3 | | 0,0262626 | 0,0222222 |
| 4 | | 0,0181818 | 0,0242424 |
| 5 | | 0,0000000 | ? |
| 6 | | 0,0161616 |  |

Data untuk proses training dan testing memiliki perbedaan jumlah dikarena pada proses training urutan (t=1) digunakan untuk awalnya saja dikarenakan di proses awal semua inputan *randdom* dan untuk urutan (t=6) digunakan untuk proses validasi didalam fase *backward.*

### Gambaran Skenario Pelatihan Metode *LSTM*

Proses pelatihan *Recurent* *Neural Network* menggunakan metode *LSTM*, yang akan menghasilkan nilai bobot akhir yang nantinya akan digunakan untuk proses uji coba. Proses pertama adalah menentukan parameter pelatihan. *Variabel* yang digunakan dalam scenario pelatihan ini yaitu: *input layer*, *hidden layer*, *output layer*, *Epoch* = 1, *learning rate* (α) = 0,2.

* + - 1. **Fase-fase setiap epcohnya**

1. **Fase pertama : Propagasi maju *(forward)***

Informasi masukan dipropagasikan ke layar tersembunyi *(hidden layer)* menggunakan fungsi *aktivasi* yang sudah ditetapkan oleh LSTM. Keluaran dari tiap unit tersembunyi akan menghasilkan keluaran, dari hasil keluaran ( tersebut dibandingkan dengan target yang harus dicapai (. Jika Selisih () menghasilkan kesalahan yang besar maka dilanjutkan ke fase *backward.*

Berikut gambaran skenario Propagasi maju *(forward)* dengan menerapakan data pada table 3 :

1. Input Layer

Didalam input layer terdapat 3 masukan dalam LSTM yaitu :

* + 1. *Samples*

*Samples* ini adalah baris dalam data, dari data *training* ditunjukkan pada tabel *3* data *training* harga *close* saham. Tabel tersebut memberikan 5 data training, maka ada 5 baris *(5* *samples).*

* + 1. *Time steps*

*Langkah waktu* adalah pengamatan masa lalu fitur, jadi data yang diproses sekarang melibatkan hasil yang dihasilkan dari tahap atau langkah sebelumnya.

1. *Features*

Fituradalah kolom dalam data, dalam sekenario ini hanya melibatkan 1 fitur saja yaitu harga close saham.

1. Inisialisasi bobot

LSTM memiliki 3 gerbang dan 1 state keadaan sel (*cell states).* Maka bobot ini berbentuk matriks dangan ukuran 4 baris dalam matriks bobot tersebut, dalam gambaran skenario ini hanya memiliki 1 fitur maka baris bobot (W) adalah 4 x 1 = 4, dan kolom bobot (W) didapat dari panjang kolom data sample yang sudah diberi bias.

Contoh data sample diurutan pertama atau langkah pertama:

Tabel 6 Data Sample

|  |  |  |
| --- | --- | --- |
| Urutan (t) | Bias | (fitur Close) |
| 1 | 1 | 0,030303030 |
| 2 | 1 | 0,0202020 |
| 3 | 1 | 0,0262626 |
| 4 | 1 | 0,0181818 |
| 5 | 1 | 0,0000000 |

Jadi kolom bobot (W) ada 3, 2 untuk panjang kolom data sample yang sudah diberi bias dan 1 untuk banyak fitur, didapatlah ukuran bobot matriks dengan ordo (4 x 3).

* bobot (W) dengan angka random :

Tabel 7 Bobot Awal

|  |  |  |  |
| --- | --- | --- | --- |
|  | -0,245714286 | 0,029262045 | 0,184398087 |
|  | 0,868020398 | -0,379580925 | 0,079506914 |
|  | -0,206444161 | -0,085253247 | 0,25112624 |
|  | 0,842874383 | 0,907722829 | -0,593738792 |

1. Persiapan block input

Pertama penggabungan data sample dengan block output sebelumnya ( untuk mencari

* Penggabungan data sample tranpose dengan block output , menggunakan persamaan (2.13)

Tabel 8 Penggabungan data sample tranpose dengan block output

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *t* |  |  |  |  |  |  |
| b | -1 | 1 | 1 | -1 | 1 | 1 |
|  | -0,0303 | 0,0202 | 0,0263 | -0,0182 | 0 | 0,017 |
|  | 0 |  | -0,093 | -0,1678 | -0,015 | -0,1352 |

sama dengan 0 dikarenakan hasil *block output* dari proses hidden layer sebelumnya belum ada, dan untuk ( adalah hasil block output dari proses hidden layer urutan ke pertama dan seterusnya.

* Perkalian antara dengan bobot ,, Menggunakan persamaan (2.16)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | 0,209028 | -0,241 | -0,262 | 0,276 | -0,248 | -0,2702 |
|  | -0,85652 | 0,8620 | 0,851 | -0,85 | 0,867 | 0,8511 |
|  | 0,244828 | -0,203 | -0,232 | 0,250 | -0,210 | -0,2418 |
|  | -0,87038 | 0,8487 | 0,922 | -0,959 | 0,852 | 0,938 |

Tabel 9 Block input untuk proses ke hidden layer t = 1

Setelah didapat block input () urutan pertama (t=1), block input ini akan menuju ke hidden layer terus menerus sampai t ke urutan 5. Gambaran hidden layer dapat dilihat di gambar 11 (Arsitektur LSTM forward Penjabaran dari Hidden Layer LSTM).

1. Hidden layer

Didalam hidden layer LSTM memiiki 3 gerbang dan dua penghubung yaitu dari semua informasi *long term* ( dan *short* term (block output ). Jadi dalam proses *hidden layer* ini menghitung .

Ditemukan dari proses *block input* ) menghasilkan yang ditunjukan pada tabel 9.

**Langkah 1 Menghitung Gerbang lupa dengan persamaan (2.3) :**

=

melakukan proses sama sesuai dengan rumus diatas.

Tabel 10 Hasil proses **Gerbang lupa (**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 0,55207 | 0,449 | 0,4422 | 0,5622 | 0,44765 |

**Langkah 2** Gerbang pembuatan vektor baru dengan persamaan (2.6), (2.8)

* =

melakukan proses sama sesuai dengan persamaan diatas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 0,29807 | 0,7031 | 0,7007 | 0,2999 | 0,7041 |

Tabel 11 Hasil proses lapisan gerbang input



melakukan proses sama sesuai dengan persamaan diatas.

Tabel 12 Hasil proses lapisan tanh candidate gate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | -0,23666 | -0,25633 | 0,269314 | -0,2435 |

Gerbang pembuatan vektor baru =

**Langkah 3** Menghitung Keadaanpenghubung *(state cell)* dengan persamaan (2.9) *:*

* = 0,071551

melakukan proses sama sesuai dengan rumus diatas.

Tabel 13 Hasil proses Keadaan penghubung (state cell)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 0,0715 | -0,1342 | -0,239 | -0,0536 | -0,1954 |

**Langkah 4** Menghitung Gerbang *output* *(output gate)* dengan persamaan (2.11) :

=

melakukan proses sama sesuai dengan rumus diatas.

Tabel 14 Hasil proses Gerbang output (output gate)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 0,295 | 0,7003 | 0,7155 | 0,27708 | 0,701 |

**Langkah 5** Menghitung *Block output* dengan persamaan (2.12) :

didapat dari table *13*

melakukan proses sama sesuai dengan rumus diatas.

Tabel 15 Hasil proses Block output (

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | -0,0934 | -0,093 | -0,1678 | -0,015 |

Fase Propagasi maju (forward) hasil dari *Block output* sama denganhasil urutan berikutnya (prediksi) dari proses *Block output .* Jadi sama dengan .

= data asli

= Prediksi dari data

=

Dari hasil *Block output* ini akan digunakan untuk proses masukan berikutnya yang ada di block input, dan seterusnya sampai banyaknya urutan informasi. Gambaran alurnya dapat dilihat pada gambar 3.1. terakhir () adalah output yang sesungguhnya atau hasil prediksi dari urutan - urutan sebelumnya.

1. **Fase kedua : Propagasi Balik *(backward)***

Berdasarkan kesalahan () dihitung *error* *output* (urutan terakhir untuk modifikasi bobot dari ke semua unit tersembunyi (*hidden layer*) hingga sampai ke unit masukan pertama dihitung.

Berikut gambaran skenario Propagasi Balik *(backward)* dengan menerapakan data pada table 5 dan dari proses lanjutan dari fase forward :

Dari fase forward, didapat . Maka dalam fase ini menghitung *error* dari dan t dimulai dari angka 5 karena 5 proses urutan paling akhir.

1. *Error Output layer* .

Dari proses forward didapat *block Output layer* atau output prediksi dimulai dari urutan paling terakhir yaitu ) dapat dilihat pada table 15 dan untukmembandingkan dengan data aktual dapat dilihat pada table 5. = 0

**Langkah 1** Menghitung Error dari data real dengan prediksi menggunakan persamaan (2.38) :

=

= data asli

= data prediksi dari data

**=**  (2.38)

Tabel 16 Hasil proses Menghitung Error dari data real dengan prediksi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 0,011462155 | 0,018878615 | 0,11186915 | 0,171721881 | 0,172162946 |

1. *Error block Output layer* .

**Langkah 2** Membandingakan antara jika jika :

Tabel 17 Perbandingan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1/0 | 1 | 1 | 1 | 1 | 0 |

Tabel 18 Perbandingan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | |  |
| 1/0 | 0 | 0 | 0 | 0 | 1 | |

**Langkah 3** Menghitung informasi *Error block Output layer* ().

*Error block Output layer*  awal sama dengan 0 karena proses dari *Error block Output layer* sebelumnya belum ada dan juga 0. sama dengan Proses *Error block Output layer* sebelumnya.

Tabel 19 Hasil proses block Output layer backward

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | 0,000 | -0,051 | -0,016 | -0,071 | -0,086 | -0,012 |

* + 1. Menghitung *error* *block Output layer dengan*

**=**  ……………………(2.39)

**=**

**=**  = -0,5

Table 20 Proses *error* *block Output layer dengan*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | |  |
|  | -0,5000 | -0,5507 | -0,5159 | -0,5708 | -0,0860 | |

* + 1. Menghitung error *block Output layer dengan <*

**=**  ……………………………(2.40)

**=**

**=**  = -0,5

Tabel 21 Proses error block Output layer dengan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | -0,5000 | -0,5507 | -0,5159 | -0,5708 | 0,4140 |

melakukan proses sama sesuai dengan rumus diatas.

1. Pada proses mencari pada forward ditemukan persamaan (2,12) maka langkah selanjutnya menghitung

**Langkah 4** Menghitung informasi *Error Output gate* dengan persamaan (2.18)

t menunjukkan pada urutan dimulai dari 5

…………………………………………..(2.18)

= dapat dilihat pada *Table* *13,*

-0,5 tanh(-0,1954) = 0,09647763

Tabel 22 Proses *error Error Output gate*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 0,096 | 0,029 | 0,121 | 0,076 | 0,030 |

melakukan proses sama sesuai dengan rumus diatas.

Langkah 4 dan langkah 5 adalah proses *Output gate*

**Langkah 5** Menghitung informasi Error Output gate () dangan persamaan (2.23).

Tabel 23 Proses Error Output gate sebelumnya ()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | 0 | -0,337 | -0,303 | -0,519 | -0,622 |

………………(2.19)

-0,337411626

Tabel 24 Proses Menghitung Error cell state ()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | -0,337 | -0,303 | -0,519 | -0,622 | -0,158 |

melakukan proses sama sesuai dengan rumus diatas.

1. Pada proses forward diberikanpersamaan (2.9) dan ditemukan

: *candidate gate*

**Langkah 6** Hitung informasi *error* gerbang input (*input gate*) :

…………………….............................................(2.20)

-0,337 -0,1954 = 0,08214642

**Langkah 7** Hitung informasi *error* gerbang *candidate connecting state*

…………………………………………………(2.21)

-0,337 0,7041 = -0,237567466

Langkah 6 dan langkah 7 adalah proses *write gate*

**Langkah 8** Hitung informasi *error* gerbang *forget gate*

……………………………………………….(2.22)

-0,337 -0,0536 = 0,018080133

Langkah 8 adalah proses *Keep gate*

**Langkah 9** Hitung informasi error Keadaan penghubung sebelumnya

………………………………………………(2.23)

-0,337 0,44765= -0,151042409

Langkah 9 adalah proses *Update cell state* dalam fase *backward*

Table 25 proses Menghitung

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t |  |  |  |  |  |
|  | 0,082 | -0,082 | 0,133 | 0,147 | -0,038 |
|  | -0,238 | -0,091 | -0,364 | -0,437 | -0,047 |
|  | 0,018 | 0,072 | 0,070 | -0,045 | 0,000 |
|  | -0,151 | -0,170 | -0,230 | -0,280 | -0,087 |

Untukdalam mencari dilakukan proses sama sesuai dengan persamaan diatas. nantinya akan dibawa ke *hidden layer* sebelumnya dalam *table 23.*

1. Pada proses forward diberikanpersamaan (2.24) dan ditemukan maka menghitung dan menghasilkan :

**Langkah 10** Hitung *error block input* *candidate gate*

(Terdapat pada *table 9*)

…..……………………………(2.24)

= - 0,223486118

**Langkah 11** Hitung *error block input input gate*

…………………………………….(2.25)

= 0,017115052

**Langkah 12** Hitung *error block input forget gate*

...…………………………………(2.26)

= 0,004470485

**Langkah 13** Hitung error block input *output gate*

...……………………………….. (2.27)

= 0,020224728

**Langkah 14** Menggabungkan langkah 10, 11, 12,13 menjadi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| T |  |  |  |  |  |
|  | -0,223 | -0,223 | -0,223 | -0,223 | -0,223 |
|  | 0,017 | 0,017 | 0,017 | 0,017 | 0,017 |
|  | 0,004 | 0,004 | 0,004 | 0,004 | 0,004 |
|  | 0,020 | 0,020 | 0,020 | 0,020 | 0,020 |

Tabel 26 Hasil proses untuk block input

Dari langkah 14 didapat *block input* ().

Untukdalam mencari dilakukan proses sama sesuai dengan rumus diatas sampai .

1. Dari tahapdiberikan persamaan (2.17) dan ditemukan dari langkah sebelumnya untuk dicari .

**Langkah 15**

Tahap untuk menghitung

……………………………………………….(2.29)

..………………………………………………(2.31)

=

=

Tabel 27 Hasil proses untuk menghasilkan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *t* |  |  |  |  |  |
| b | 0,0859 | 0,0071 | 0,1250 | 0,1439 | 0,0092 |
|  | 0,0049 | 0,0079 | 0,0003 | -0,0083 | 0,0073 |
|  | -0,0507 | -0,0159 | -0,0708 | -0,0860 | -0,0125 |

Dari proses ini didapat akan menghubungkan ke layer sebelmunya dan tambahkan ke tabel 19 pada kolom , proses ini mengulangi terus sampai awal

1. Menghitung Derivatif Bobot

Menggunakan aturan rantai konvensional untuk menghitung turunan dari E sehubungan dengan semua bobot. Dalam kondisi tertentu, ini bisa menjadi bobot pendekatan.[8]

……………………………………………………(2.33)

…(2.34)

Tabel 28 Hasil Derivatif Bobot

|  |  |  |  |
| --- | --- | --- | --- |
| t |  | | |
|  | -0,22349 | - 0,00361 | 0,030226 |
| 0,017115 | 0,000276607 | -0,00231 |
| 0,00447 | 7,22503E-05 | -0,0006 |
| 0,020225 | 0,000326864 | -0,00274 |
|  | -0,08433 | 0 | 0,001251 |
| -0,01714 | 0 | 0,000254 |
| 0,017834 | 0 | -0,00026 |
| 0,005906 | 0 | -8,8E-05 |
|  | -0,33997 | -0,006181 | 0,057047 |
| 0,027917 | 0,000507579 | -0,00468 |
| 0,017195 | 0,000312629 | -0,00289 |
| 0,024631 | 0,000447845 | -0,00413 |
|  | -0,41299 | -0,008343 | -0,00871 |
| 0,030742 | 0,000621048 | 0,000648 |
| -0,01102 | -0,000222 | -0,00023 |
| 0,015985 | 0,000322927 | 0,000337 |
|  | -0,0444 | -0,001345 | 0 |
| -0,00794 | -0,0002405 | 0 |
| 0 | 0 | 0 |
| 0,006153 | 0,000186448 | 0 |

Bobot ini akan di proses penjumlahan untuk mendapatkan bobot final dengan rumus :

……………………………...(2.35)

Tabel 29 Modifikasi bobot untuk proses optimasi

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | (lama) | | | |  | | | |  | | | |
|  | -0,22 | - 0,004 | 0,03 | 0 | | 0 | 0 | -0,22 | | - 0,004 | 0,03 |
| 0,017 | 0,0003 | -0,002 | 0 | | 0 | 0 | 0,017 | | 0,0003 | -0,002 |
| 0,004 | 7,2E-05 | -0,0006 | 0 | | 0 | 0 | 0,004 | | 7,2E-05 | -0,0006 |
| 0,020 | 0,0003 | -0,003 | 0 | | 0 | 0 | 0,020 | | 0,0003 | -0,003 |
|  | -0,08 | 0 | 0,001 | -0,223 | | -0,00361 | 0,03 | -0,308 | | -0,0036 | 0,0315 |
| -0,017 | 0 | 0,0002 | 0,017 | | 0,000277 | -0,002 | -2,9E-05 | | 0,00028 | -0,002 |
| 0,018 | 0 | -0,0003 | 0,004 | | 7,2E-05 | -0,0006 | 0,022 | | 7,2E-05 | -0,0009 |
| 0,006 | 0 | -8,8E-05 | 0,020 | | 0,0003 | -0,003 | 0,026 | | 0,0003 | -0,0028 |
|  | -0,34 | -0,006 | 0,06 | -0,308 | | -0,004 | 0,03 | -0,65 | | -0,01 | 0,09 |
| 0,028 | 0,0005 | -0,005 | -2,9E-05 | | 0,0003 | -0,002 | 0,028 | | 0,0008 | -0,007 |
| 0,017 | 0,0003 | -0,003 | 0,02 | | 7,2E-05 | -0,0009 | 0,04 | | 0,0004 | -0,0037 |
| 0,025 | 0,0004 | -0,004 | 0,026 | | 0,0003 | -0,003 | 0,0508 | | 0,00077 | -0,007 |
|  | -0,413 | -0,008 | -0,009 | -0,648 | | -0,01 | 0,09 | -1,060 | | -0,018 | 0,08 |
| 0,031 | 0,0006 | 0,0006 | 0,028 | | 0,0008 | -0,007 | 0,058 | | 0,0014 | -0,006 |
| -0,01 | -0,0002 | -0,00023 | 0,039 | | 0,0004 | -0,0037 | 0,028 | | 0,00016 | -0,004 |
| 0,016 | 0,00032 | 0,00034 | 0,0508 | | 0,0008 | -0,007 | 0,067 | | 0,0011 | -0,0067 |
|  | -0,044 | -0,001 | 0 | -1,0608 | | -0,019 | 0,08 | -1,105 | | -0,0195 | 0,08 |
| -0,008 | -0,0002 | 0 | 0,059 | | 0,0014 | -0,0061 | 0,051 | | 0,0012 | -0,0061 |
| 0 | 0 | 0 | 0,0285 | | 0,0002 | -0,004 | 0,0285 | | 0,0002 | -0,004 |
| 0,006 | 0,00019 | 0 | 0,067 | | 0,0011 | -0,007 | 0,0729 | | 0,0013 | -0,007 |

1. **Fase ketiga : Perubahan bobot optiamsi bobot *(final* bobot*)***

Setelah semua *error* *output* (dihitung, dan semua bobot dimodifikasi bersamaan maka bobot modifikasi tersebut diupdate dengan metode (*gradient descent*). Dan didapat final bobot yang akan digunakan untuk proses pengujian ke data pegujian.

= 0,2

…………………………(2.37)

Tabel 30 Bobot yang sudah di update ()

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t |  | | | |  | | |  | | |
|  | -0,24571 | 0,0293 | 0,1844 | -0,221 | | -0,0039 | 0,016 | -0,025 | 0,0133 | 0,1844 |
|  | 0,8680 | -0,3796 | 0,0795 | 0,010 | | 0,0002 | -0,0012 | 0,85789 | -0,37837 | 0,0795 |
|  | -0,2064 | -0,085 | 0,251 | 0,0057 | | 3,2E-05 | -0,0008 | -0,212 | -0,084 | 0,251 |
|  | 0,84287 | 0,908 | -0,594 | 0,0146 | | 0,00026 | -0,0013 | 0,828 | 0,9090 | -0,5937 |

Dari pelatihan dengan epoch 1 dan learning\_rate 0,2 dengan data *training* yang ditunjukakn di *table 3* menghasilkan *update* bobot baru yang ada di tabel 30 , bobot inilah yang akan dilanjutkan ke urutan berikutnya. Operasi-operasi ini berproses sama dengan menggunakan bobot-bobot akhir hasil pengolahan data pertama ini sebagai bobot-bobot awalnya. Proses ini dilakukan berulang sampai pada maksimum *epoch.*

### Gambaran Skenario Pengujian Metode *LSTM*

Pada pengujian ini dilakukan hanya fase *forward* saja. Dimana menggunakan bobot yang sudah didapat pada proses *training* sebelumnya.

Berikut gambaran skenario Propagasi maju *(forward)* dengan menerapakan data pada table 4 (data *testing* ) :

1. Input Layer
2. Inisialisasi bobot yang didapat dari proses training sebelumnya.

Tabel 31 Data sample testing

|  |  |  |
| --- | --- | --- |
| Urutan (t) | bias | (fitur Close) |
| 1 | 1 | 0,0161616 |
| 2 | 1 | 0,0262626 |
| 3 | 1 | 0,0222222 |
| 4 | 1 | 0,0242424 |
| 5 | 1 | ? |

* Didapat bobot dari proses sebelumnya, yaitu proses *training* :

Tabel 32 Bobot baru

|  |  |  |  |
| --- | --- | --- | --- |
|  | -0,025 | 0,0133 | 0,1844 |
|  | 0,85789 | -0,37837 | 0,0795 |
|  | -0,212 | -0,084 | 0,251 |
|  | 0,828 | 0,9090 | -0,5937 |

1. Persiapan block input

* Penggabungan data sample tranpose dengan block output dengan menggunakan persamaan (2.13)

Tabel 33 Penggabungan data sample tranpose dengan block output

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *t* |  |  |  |  |
| b | 1 | 1 | 1 | 1 |
|  | 0,0161 | 0,0262 | 0,022 | 0,0242 |
|  | 0,0000 | -0,012 | -0,019 | -0,022 |

sama dengan 0 dikarenakan hasil *block output* dari proses hidden layer sebelumnya belum ada, dan untuk ( adalah hasil block output dari proses hidden layer urutan sebelumnya dan seterusnya.

* Perkalian antara dengan bobot ,, dengan menggunakan persamaan (2.15)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | -0,0245 | -0,0265 | -0,027 | -0,0284 | -0,0245 |
|  | 0,8518 | 0,847 | 0,85 | 0,847 | 0,852 |
|  | -0,2135 | -0,217 | -0,219 | -0,22 | -0,2135 |
|  | 0,843 | 0,86 | 0,859 | 0,8633 | 0,843 |

Tabel 34 Block input untuk proses ke hidden layer t = 1

Setelah didapat block input () urutan pertama (t=1), block input ini akan menuju ke *hidden layer* terus menerus sampai t ke urutan 5. Gambaran *hidden layer* dapat dilihat di gambar 11 (Arsitektur LSTM forward Penjabaran dari Hidden Layer LSTM

1. Hidden layer

**Langkah 1 Gerbang lupa *(forget gate)*:**

…………………………………………………(2.3)

=

melakukan proses sama sesuai dengan rumus diatas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0,4468 | 0,446 | 0,445 | 0,4453 |

Tabel 35 Hasil proses Gerbang lupa ()

**Langkah 2 Gerbang pembuatan vector baru (input gate)**:

* ………………………………………..……….…(2.6)

=

melakukan proses sama sesuai dengan persamaan diatas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0,7009 | 0,69 | 0,7 | 0,69 |

Tabel 36 Hasil proses lapisan gerbang input

1. ……………………………………………..(2.8)

melakukan proses sama sesuai dengan persamaan diatas.

Tabel 37 Hasil proses lapisan tanh candidate gate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | -0,23666 | -0,25633 | 0,269314 | -0,2435 |

Gerbang pembuatan vektor baru =

**Langkah 3** Keadaanpenghubung *(state cell) :*

……………………………………… (2.9)

= -0,017143588

melakukan proses sama sesuai dengan persamaan diatas.

Tabel 38 Hasil proses Keadaan penghubung (state cell)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | -0,0171 | -0,0265 | -0,278 | -0,028 |

**Langkah 4** Gerbang *output* *(output gate) :*

…………………………………………………. (2.11)

=

melakukan proses sama sesuai dengan persamaan diatas.

Tabel 39 Hasil proses Gerbang output (output gate)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0,699 | 0,7025 | 0,7025 | 0,70335 |

**Langkah 5** *Block output*

……………………………………………(2,12)

didapat dari table *13*

melakukan proses sama sesuai dengan persamaan diatas.

Tabel 40 hasil proses Block output (

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | -0,0198 | -0,0184 | -0,022 | -0,0237 |

Fase Propagasi maju (forward) hasil dari *Block output* sama denganhasil urutan berikutnya (prediksi) dari proses *Block output .* Jadi sama dengan .

1. Denorlamalisasi (menjadikan hasil prediksi)

…………………………………………(2.41)

= -0,0237

\* 12375 +12000

11706,59

Jadi hasil prediksi hari ke lima dari proses prediksi () diatas adalah 11706,59 .

1. Setelah itu akan dilakukan perbandingan antara data sebenarnya dengan data hasil prediksi, sehingga dapat dihitung *error*.

dapat dilihat di table 5

E =

E = 12000 - 11706,59

E = 293,41

Pada Gambaran Skenario Metode LSTM ini hanya memakai 10 data, data training 5 dan data testing 5 dan menghasilkan 1 output prediksi.

## Skenario Uji Coba dan Evaluasi

Tahap ini akan dilakukan uji coba Pada penelitihan 1500 data yang dibagi menjadi 3 bagian dengan cara *Multiple Train-Test Split,* yangdibagi menjadi 3.

Tabel 41 Uji coba dengan pembagian data

|  |  |  |
| --- | --- | --- |
| Ujicoba | Trainig | Testing |
| 1 | 250 DATA | 250 DATA |
| 2 | 500 DATA | 500 DATA |
| 3 | 1000 DATA | 500 DATA |