

SKRIPSI

Rachmad Agung Pambudi

160411100032



**PROGRAM STUDI S1 INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
BANGKALAN
2020**

SKRIPSI

**Diajukan untuk Memenuhi Persyaratan
Penyelesaian Studi Strata Satu (S1)
dan Memperoleh Gelar Sarjana Komputer (S.Kom)
di Universitas Trunojoyo Madura**

**Rachmad Agung Pambudi
160411100032**

**PROGRAM STUDI S1 INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
BANGKALAN
2020**

HALAMAN PENGESAHAN

PENERAPAN ALGORITMA LSTM (*Long-Short Term Memory*) UNTUK PREDIKSI HARGA SAHAM

Skripsi ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer (S.Kom) di Universitas Trunojoyo Madura

Oleh :

Nama : Rachmad Agung Pambudi

NIM : 160411100032

Disetujui oleh TIM Penguji Skripsi :

Tanggal Sidang:

x - x - 2020

Eka Mala Sari Rochman, S.Kom., M.Kom

NIP. 19841104 200812 2003

(Pembimbing I)

Sri Herawati, S.Kom., M.Kom

NIP. 19830828 200812 2002

(Pembimbing II)

Yoga Dwitya Pramudita, S.Kom., M.Cs

NIP. 19840413 200812 1 002

(Penguji I)

Husni, S.Kom., M.Kom

NIP. 19790722 200312 1 001

(Penguji II)

Muhammad Ali Syakur, S.Si., M.T

NIP. 19690115 200312 1 001

(Penguji III)

Bangkalan, x – x – 2020

Mengetahui,

Ketua Jurusan Teknik Informatika

Sigit Susanto Putro, S.Kom., M.Kom

NIP. 19790313 200604 1 002

HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul:

“PENERAPAN ALGORITMA LSTM (*Long-Short Term Memory*) UNTUK PREDIKSI HARGA SAHAM ”

1. Adalah asli, bukan merupakan karya pihak lain serta belum pernah diajukan untuk mendapatkan gelar akademik Sarjana Komputer baik di Universitas Trunojoyo Madura maupun di Perguruan Tinggi yang lain di Indonesia.
2. Tidak terdapat karya atau pendapat pihak lain yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis telah diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila dikemudian hari terbukti skripsi ini sebagian atau seluruhnya merupakan hasil plagiasi atau terdapat hal-hal yang tidak sesuai dengan pernyataan diatas, maka saya sanggup menerima sanksi akademis yang berlaku dengan segala akibat hukumannya sesuai peraturan Universitas Trunojoyo Madura dan atau peraturan perundang-undangan yang berlaku.

Bangkalan, x – x – 2020

Yang menyatakan,

Rachmad Agung Pambudi
NIM. 160411100032

HALAMAN PERSEMBAHAN

PENERAPAN ALGORITMA LSTM (*Long-Short Term Memory*)

UNTUK PREDIKSI HARGA SAHAM

Rachmad Agung Pambudi
(160411100032)

Pembimbing I : Eka Mala Sari Rochman, S.Kom.,
M.Kom
Pembimbing II : Sri Herawati, S.Kom., M.Kom

ABSTRAK

Saham merupakan salah satu pilihan dalam investasi yang menarik karena dapat menghasilkan keuntungan yang besar, jika dapat membaca pergerakan perubahan harga saham. Bagi pemula untuk memulai berinvestasi saham memiliki resiko kerugian yang sangat besar dikarenakan belum bisa membaca pergerakan harga saham. Untuk meminimalkan resiko kerugian diperlukan belajar membaca pergerakan harga saham dan perkembangan pasar modal yang merupakan indikator dalam masuk didunia investasi saham. Dengan perkembangan teknologi kecerdasan buatan dan pemrosesan prediksi saat ini, pergerakan harga saham dapat diidentifikasi dengan cara otomatis secara perhitungan matematis yang sangat kompleks. *Deep Learning* merupakan salah satu teknologi kecerdasan buatan yang memiliki akurasi pengenalan yang tinggi dengan data yang sangat banyak. Penelitian ini menggunakan analisis harga saham sebelumnya dalam suatu perusahaan, dengan teknik *Deep learning*, *Recurrent Neural Networks (RNN)* modul pemrosesan *Long-Short Term Memory (LSTM)* untuk melakukan prediksi harga saham. Fitur yang digunakan dalam pemrosesan prediksi yaitu Data harga saham PT Semen Indonesia (Persero) Tbk saat harga penutupan (harga *closing*) dari tanggal (09/01/2014) sampai (09/01/2020), sebanyak 1500 data. Diharapkan dengan metode *Long-Short Term Memory (LSTM)* bisa memberikan performa terbaik untuk memprediksi harga saham dalam seminggu.

Kata kunci: *Deep Learning*, *Long-Short Term Memory (LSTM)*, *Recurrent Neural Networks (RNN)*, Prediksi harga saham.

KATA PENGANTAR

Bismillahirrahmanirrahim, Alhamdulillah puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya yang sangat besar sehingga pada akhirnya penulis mampu menyelesaikan skripsi, dengan judul **“Penerapan Algoritma LSTM (Long-Short Term Memory) untuk Prediksi Harga Saham”**. Dengan terselesaikannya laporan skripsi ini, penulis mengucapkan banyak terima kasih sebesar-besarnya kepada:

1. Allah SWT.
2. Kedua orang tua penulis yang telah memberikan dukungan, bimbingan dan do’a yang tiada henti sehingga penulis dapat menyelesaikan laporan skripsi ini.
3. Eka Mala Sari Rochman, S.Kom., M.Kom. selaku dosen pembimbing I dan Sri Herawati, S.Kom., M.Kom. selaku dosen pembimbing II. Terima kasih telah banyak meluangkan waktu untuk membimbing dan memberikan arahan kepada penulis selama penelitian skripsi.
4. Bapak dan Ibu Dosen Fakultas Teknik, khususnya Prodi Informatika Universitas Trunojoyo Madura yang telah membagikan ilmunya kepada penulis selama masa perkuliahan.
5. Teman-teman kuliah Teknik Informatika angkatan 2016 (IMAGINE), terima kasih telah banyak membantu dalam kelancaran penyelesaian skripsi dan momen bahagia yang kita ciptakan bersama selama empat tahun.
6. Teman-teman main dan khususnya Nadila Hidayanti, terima kasih untuk waktunya telah menemani penulis saat mengerjakan skripsi bersama-sama.
7. Pihak-pihak lain yang tidak sempat penulis sebutkan, yang telah membantu dalam kelancaran pengerjaan skripsi ini.

Penulis menyadari bahwa pengerjaan skripsi ini masih jauh dari kesempurnaan dan terdapat kekurangan didalamnya. Maka penulis mengharapkan adanya kritik dan saran yang membangun dari pembaca demi kesempurnaan skripsi ini.

Demikian akhirnya penulis mengucapkan banyak terima kasih dan mohon maaf apabila terdapat kesalahan baik tutur kata dalam pengetikan, penulisan, penyampaian maupun hasil pengamatan. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pembaca dan kita semua. Aamiin

Bangkalan, x – x – 2020

Yang menyatakan,

Rachmad Agung Pambudi
NIM. 160411100032

DAFTAR ISI

SKRIPSI	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN ORISINALITAS	iv
HALAMAN PERSEMBAHAN	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.2.1 Permasalahan	2
1.2.2 Metode Usulan	2
1.2.3 Pertanyaan Penelitian	2
1.3 Tujuan dan Manfaat	2
1.3.1 Tujuan Penelitian	2
1.3.2 Manfaat Penelitian	2
1.4 Batasan-batasan	3
1.5 Sistematika Proposal	3
BAB II KAJIAN PUSTAKA	4
2.1 Pengertian Saham	4
2.1.1 Saham PT Semen Indonesia (Persero) Tbk (SMGR)	4
2.2 <i>Recurrent neural network</i> (RNN)	5
2.3 <i>Long Short Term Memory</i> (LSTM)	9
2.3.1 Algoritma pelatihan <i>Long Short Term Memory Networks</i>	12
2.3.2 Feedforward	14
2.3.3 Backpropagation Through Time (BPTT)	15
2.3.4 Mencari Bobot dari proses backpropagation urutan (sequence)	18
2.3.5 Optimasi untuk update bobot Gradient Descent	19
2.4 Penelitian Terkait	21
BAB III METODE USULAN	23
3.1 Desain Rancangan	23
3.1.1 Rancangan Arsitektur LSTM	23
3.1.2 Diagram I-P-O	25

3.1.3 Diagram Pengembangan.....	26
3.2 Analisa Kebutuhan	28
3.2.1 Kebutuhan <i>Hardware</i>	28
3.2.2 Kebutuhan <i>Software</i>	28
3.3 Analisa Input	28
3.4 Analisa Proses	28
3.5 Analisa Output	29
3.6 Desain Rancangan Sistem.....	29
3.6.1 Flowchart Diagram	29
3.6.1 Flowchart Multiple Train-Test Split.....	32
3.6.2 Flowchart Tahap Pelatihan LSTM.....	32
3.6.3 Flowchart Tahap Pengujian LSTM.....	34
3.7 Rancangan Skenario Metode <i>LSTM</i>	35
3.7.1 Gambaran Skenario Pelatihan Metode <i>LSTM</i>	36
3.7.2 Gambaran Skenario Pengujian Metode <i>LSTM</i>	49
3.8 Skenario Uji Coba dan Evaluasi.....	52
BAB IV IMPLEMENTASI SISTEM	53
4.1 Lingkungan Uji Coba.....	53
4.2 <i>Source Code</i> Objek LSTM.....	54
4.3 Graphical User Interface (GUI)	58
4.3.1 Tampilan Aplikasi.....	58
4.3.3 Tampilan fitur	59
4.4 Uji Coba Sistem	64
4.4.1 Uji Coba Skenario.....	64
BAB V PENUTUP.....	69
5.1. Kesimpulan.....	69
5.2. Saran.....	69
DAFTAR PUSTAKA	53

DAFTAR GAMBAR

Gambar 2.1 Arsitektur pada RNN.....	5
Gambar 2.2. Hidden layer pada RNN	5
Gambar 2.3 Memori pada LSTM.....	6
Gambar 2.4 Arsitektur LSTM.....	9
Gambar 2.5 Status sel LSTM.....	9
Gambar 2.6 Gerbang LSTM	10
Gambar 2.7 Penjagaan gerbang (Keep gate) ft LSTM	10
Gambar 2.8 Menambahkan informasi baru (write gate) LSTM	11
Gambar 2.9 Update cell state LSTM	11
Gambar 2.10 Membaca keputusan (Output) LSTM	12
Gambar 3.1 Arsitektur LSTM forward	23
Gambar 3.2 Penjabaran dari Hidden Layer LSTM	23
Gambar 3.3 Arsitektur LSTM Backward.....	24
Gambar 3.4 Penjabaran dari Hidden Layer LSTM Backward.....	24
Gambar 3.5 Diagram IPO Arsitektur Sistem	25
Gambar 3.6 Diagram Pengembangan	26
Gambar 3.7 Flowchart Sistem Keseluruhan	30
Gambar 3.8 Multiple Train-Test Split.....	32
Gambar 3.9 Flowchart pelatihan LSTM	33
Gambar 3.10 Flowchart pengujian LSTM	34
Gambar 4.1 Tampilan Awal Aplikasi	58
Gambar 4.2 Tampilan Menu fitur – fitur	58
Gambar 4.3 Tampilan tabel dan grafik full data close saham.....	59
Gambar 4.4 Tampilan Training data	60
Gambar 4.5 Tampilan Testing Data.....	61
Gambar 4.6. Tampilan tabel dan grafik prediksi data close saham	62
Gambar 4.7. Tampilan full Aplikasi	63
Gambar 4.8 Tampilan hasil testing data scenario 1	65
Gambar 4.9. Tampilan hasil testing data scenario 2	67
Gambar 4.10 Tampilan hasil testing data scenario 3	68
Gambar 6.1 Tampilan hasil testing data scenario 1 terbaik	56
Gambar 6.2. Tampilan hasil testing data scenario 2 terbaik	57

DAFTAR TABEL

Table 1. Tabel Penelitian Terkait	21
Tabel 2 Data Harga Saham PT Semen Indonesia (Persero) Tbk	35
Tabel 3 Data Training Harga close Saham	35
Tabel 4 Data Testing Harga Close Saham	36
Tabel 5 Data Training dan data Testing Harga close Saham dinormalisasi	36
Tabel 6 Data Sample	37
Tabel 7 Bobot Awal	38
Tabel 8 Penggabungan data sample tranpose dengan block output ($ht - 1$).....	38
Tabel 9 Block input untuk proses ke hidden layer $t = 1$	38
Tabel 10 Hasil proses Gerbang lupa (ft).....	39
Tabel 11 Hasil proses lapisan gerbang input (it)	39
Tabel 12 Hasil proses lapisan tanh candidate gate.....	39
Tabel 13 Hasil proses Keadaan penghubung (state cell)	40
Tabel 14 Hasil proses Gerbang output (output gate)	40
Tabel 15 Hasil proses Block output (ht)	40
Tabel 16 Hasil proses Menghitung Error dari data real dengan prediksi.....	41
Tabel 17 Perbandingan $yt + 1$ lebih besar dari ht	41
Tabel 18 Perbandingan $yt + 1$ lebih besar dari ht	42
Tabel 19 Hasil proses block Output layer ($\delta ht + 1$) backward	42
Table 20 Proses <i>error block Output layer δht dengan $yt + 1 > ht$</i>	42
Tabel 21 Proses <i>error block Output layer δht dengan $yt + 1 < ht$</i>	42
Tabel 22 Proses <i>error Error Output gate δot</i>	43
Tabel 23 Proses Error Output gate sebelumnya ($\delta Ct + 1$).....	43
Tabel 24 Proses Menghitung Error cell state (δCt)	43
Table 25 proses Menghitung $\delta it, \delta at, \delta ft, \delta Ct - 1$	44
Tabel 26 Hasil proses $\delta zat, \delta zit, \delta zft, \delta zot$ untuk block input δzt	45
Tabel 27 Hasil proses δIt untuk menghasilkan $\delta ht - 1$	46
Tabel 28 Hasil Derivatif Bobot δWt	47
Tabel 29 Modifikasi bobot untuk proses optimasi.....	48
Tabel 30 Bobot yang sudah di update ($Wt(Baru)$)	48
Tabel 31 Data sample testing	49
Tabel 32 Bobot baru $Wt baru$	49
Tabel 33 Penggabungan data sample tranpose dengan block output ($ht - 1$).....	49
Tabel 34 Block input untuk proses ke hidden layer $t = 1$	50
Tabel 35 Hasil proses Gerbang lupa (ft).....	50
Tabel 36 Hasil proses lapisan gerbang input (it)	50
Tabel 37 Hasil proses lapisan tanh candidate gate.....	51
Tabel 38 Hasil proses Keadaan penghubung (state cell)	51
Tabel 39 Hasil proses Gerbang output (output gate)	51
Tabel 40 hasil proses Block output (ht)	52
Tabel 41 Uji coba dengan pembagian data	52
Table 42 Skenario Uji Coba.....	64
Table 43 Data training dan data testing skenario 1	64
Table 44 Perubahan Uji Coba dengan perubahan nilai epoch pada data	64

Table 45 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data	65
Table 46 Hasil terbaik Skenario 1	65
Table 47 Data training dan data testing skenario 2	66
Table 48 Perubahan Uji Coba dengan perubahan nilai epoch pada data	66
Table 49 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data	66
Table 50 Hasil terbaik Skenario 1	66
Table 51 Data training dan data testing skenario 3	67
Table 52 Perubahan Uji Coba dengan perubahan nilai Epoch pada data	67
Table 53 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data	68
Table 54 Hasil terbaik Skenario 3	68

BAB I PENDAHULUAN

1.1 Latar Belakang

Pada era digital saat ini, Investasi saham di pasar modal pada setiap negara merupakan aset yang sangat penting bagi setiap perusahaan didunia. Sebab secara langsung maupun tidak, investor dari seluruh dunia dapat memberikan dampak terhadap ekonomi pada negara tempat berinvestasi. Saham adalah surat berharga yang dikeluarkan oleh perusahaan BUMN ataupun Swasta yang berbentuk Perseroan Terbatas (PT). Pemilik saham juga memiliki hak untuk memiliki sebagian dari perusahaan. Banyak masyarakat dari yang remaja, dewasa, hingga lansia dapat membeli atau menanamkan modalnya kedalam pasar modal untuk mengharapkan keuntungan. Hal ini senada dengan pendapat Widoatmodjo yaitu ada beberapa keuntungan dalam membeli saham adalah Capital Gain, deviden, dan saham juga dapat dijaminkan ke bank sebagai agunan untuk memperoleh kredit [1].

Prediksi (peramalan) adalah proses dalam menduga atau memperkirakan kejadian yang terjadi diwaktu mendatang dengan memanfaatkan informasi atau data yang relevan pada waktu sebelumnya (historis) melalui metode ilmiah [2].

Kegiatan untuk memprediksi harga saham banyak dilakukan dengan berbagai teknik pemodelan dan penerapan metode tertentu. Pada penelitian sebelumnya telah dilakukan perbandingan metode untuk memprediksi harga saham sektor teknologi Amerika dengan menggunakan metode Backpropagation, SVM, dan LSTM (*Long-Short Term Memory*) untuk backpropagation menghasilkan ketepatan 68.649% dengan epochs 10 dan 67,434% dengan epochs 100, untuk SVM menghasilkan ketepatan 66.9823% dengan epochs 10 dan 67.1212% dengan epochs 100, LSTM menghasilkan ketepatan 68.51635% dengan epochs 10 dan 69.04171% dengan epochs 100 [3].

Long Short Term Memory networks (LSTM) merupakan sebuah evolusi dari arsitektur RNN, dimana pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber (1997) [4]. RNN memiliki kekurangan yaitu dapat dilihat pada inputan yang memiliki rentang informasi yang sangat besar sehingga ketika block output (h) memerlukan informasi yang relevan maka RNN tidak dapat menghubungkan informasi karena memori lama yang tersimpan akan tidak berguna dengan seiringnya berjalannya waktu karena akan tergantikan dengan memori baru, permasalahan ini ditemukan oleh *Bengio, et al.* (1994). Berbeda dengan RNN, LSTM tidak memiliki kekurangan

tersebut karena LSTM (*Long-Short Term Memory*) dapat mengatur memori di setiap masukannya dengan menggunakan memory cells dan gate units [5].

Penelitian ini akan menerapkan metode LSTM (*Long-Short Term Memory*) karena pada penelitian sebelumnya mendapatkan hasil yang baik dalam proses prediksi. Pada penelitian ini menggunakan metode LSTM dengan objek harga penutupan PT Semen Indonesia (Persero) Tbk, dengan tujuan memprediksi harga penutupan dalam jangka waktu 5 hari berikutnya.

1.2 Perumusan Masalah

1.2.1 Permasalahan

Permasalahan yang sering terjadi dalam sistem prediksi ialah keterbatasan data fitur yang mempengaruhi kinerja saham saat ini dan juga keakuratan dalam memprediksi suatu saham tertentu.

1.2.2 Metode Usulan

Penelitian ini mengusulkan Penerapan metode LSTM (*Long-Short Term Memory*) untuk prediksi harga saham.

1.2.3 Pertanyaan Penelitian

Seberapa besar keakuratan hasil prediksi harga saham terhadap nilai error menggunakan metode *Neural Network* algoritma *LSTM* ?

1.3 Tujuan dan Manfaat

1.3.1 Tujuan Penelitian

Mengetahui berapa besar akurasi nilai yang diperoleh menggunakan metode *Neural Network* algoritma *LSTM* untuk menentukan harga saham untuk periode yang akan datang.

1.3.2 Manfaat Penelitian

Manfaat dari skripsi ini yaitu dengan adanya sistem prediksi harga saham ini, dapat mempermudah investor untuk menjual atau membeli saham agar di periode yang akan datang tidak mengalami penurunan atau kerugian.

1.4 Batasan-batasan

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Data yang akan digunakan pada penelitian ini adalah data saham disitus Alphavantage(<https://www.alphavantage.co/>) disertakannya API (Application Programming Interface) yang ada di situs tersebut dan situs pendukungnya yahoo finance.
2. Data yang akan digunakan pada penelitian ini adalah harga saham PT Semen Indonesia (Persero) Tbk saat harga penutupan dari tanggal (09/01/2014) sampai (09/01/2020), sebanyak 1500 data.
3. Data saham tidak mempertimbangkan hari libur.

1.5 Sistematika Proposal

Sistematika penulisan Proposal ini terdiri dari beberapa bab, yaitu :

BAB I PENDAHULUAN

Pada bab ini membahas tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan laporan.

BAB II KAJIAN PUSTAKA

Pada bab ini membahas tentang landasan teori dan penelitian sebelumnya.

BAB III METODE USULAN

Pada bab ini dijelaskan mengenai rancangan dari sistem yang akan dibuat setelah melakukan analisa masalah, mulai dari rancangan model sistem, serta alur dari sistem yang akan dikembangkan. penelitian dan saran mengenai metode yang digunakan.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan menjelaskan implementasi pembuatan aplikasi berdasarkan hasil rancangan yang telah di buat sebelumnya.

BAB V PENUTUP

Bab ini berisi kesimpulan dari penelitian yang disesuaikan dengan pertanyaan

BAB II KAJIAN PUSTAKA

2.1 Pengertian Saham

Saham merupakan sebuah surat berharga yang dikeluarkan oleh sebuah perusahaan, baik Swasta maupun BUMN yang mencari modal dari bursa efek dengan cara menerbitkan efek atau bisa disebut emiten. Pemilikan saham adalah juga pemilik sebagian dari perusahaan [6].

2.1.1 Saham PT Semen Indonesia (Persero) Tbk (SMGR)

Pada tanggal 17 Mei 1991 PT Semen Gresik Tbk mendapatkan persetujuan Badan Pengawas Pasar Modal (Bapepam) lewat surat bernomor S-622/PM/1991 menerbitkan prospektus IPO pada 20 Mei 1991 dengan menawarkan sebanyak 40 juta saham pada harga Rp 7.000 per saham dengan menargetkan modal segar Rp 280 miliar. Dari dana segar tersebut sebagian dialokasikan ke pembangunan pabrik semen baru di wilayah Tuban Jawa Timur. Dan sebagian saham digunakan meningkatkan kapasitas produksi dari 1 ton menjadi 1,3 juta ton pertahun.

Pada tanggal 7 Januari 2013 PT Semen Gresik (Persero) Tbk secara resmi berubah nama menjadi PT Semen Indonesia (Persero) Tbk sebagai holding dari tiga entitas anak usaha, yaitu PT Semen Gresik, PT Semen Tonasa, dan PT Semen Padang.

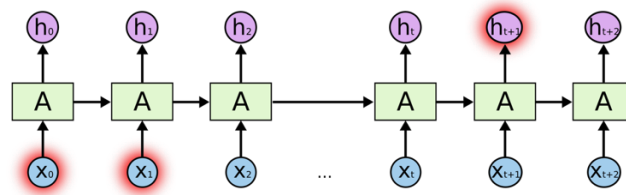
2.1.2 Harga Penutupan

Harga Penutupan (*Closing Price*) adalah harga yang muncul saat bursa tutup, harga pasar saham yang saat itu sedang berlaku akan menjadi harga penutupan untuk harga di hari itu. Harga penutupan saham hari itu juga akan menjadi acuan harga pembukaan untuk keesokan harinya.

2.1.3 Prediksi Saham

Prediksi (peramalan) adalah proses dalam menduga atau memperkirakan kejadian yang terjadi diwaktu mendatang dengan memanfaatkan informasi atau data yang relevan pada waktu sebelumnya (historis) melalui metode ilmiah [2]

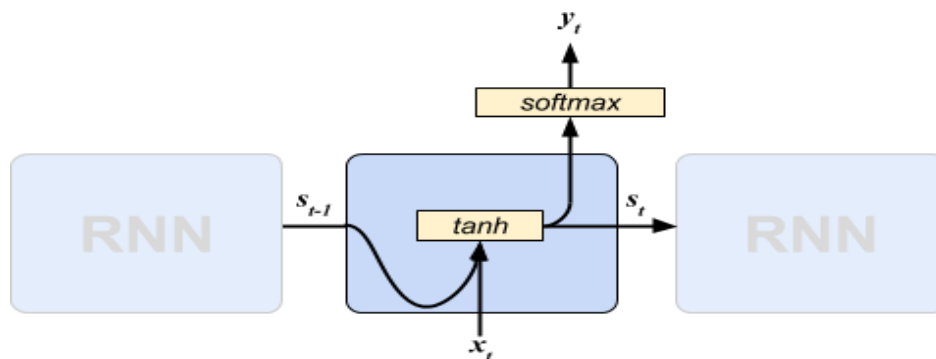
2.2 Recurrent neural network (RNN)



Gambar 2.1 Arsitektur pada RNN

Recurrent neural network (Vanilla RNN) tidak dapat belajar menghubungkan informasi jika dalam proses *training* memiliki langkah atau data yang terlalu banyak, akan menyebabkan penyimpanan lama yang tersimpan akan tidak berguna dengan seiringnya berjalannya waktu karena akan tertimpa atau tergantikan dengan memori baru (*Vanishing Gradient Problem*), permasalahan ini ditemukan oleh Bengio, et al. (1994). Berbeda dengan RNN, *Long Short Term Memory (LSTM)* tidak memiliki kekurangan tersebut (*Vanishing Gradient Problem*), karena *Long Short Term Memory (LSTM)* dapat mengatur memori pada setiap masukannya dengan menggunakan *memory cells* dan *gate units* [5]. Jenis jaringan saraf ini berguna ketika mengingat urutan data yang panjang dan tidak bergantung pada dataset jendela yang tertinggal sebagai input [3].

Pada pemrosesan dalam suatu modul *Recurrent neural network (RNN)* memiliki lapis tanh saja dan jika output dikehendaki maka akan dilewatkan dengan fungsi aktivasi softmax untuk mendapatkan output. Seperti Gambar 2

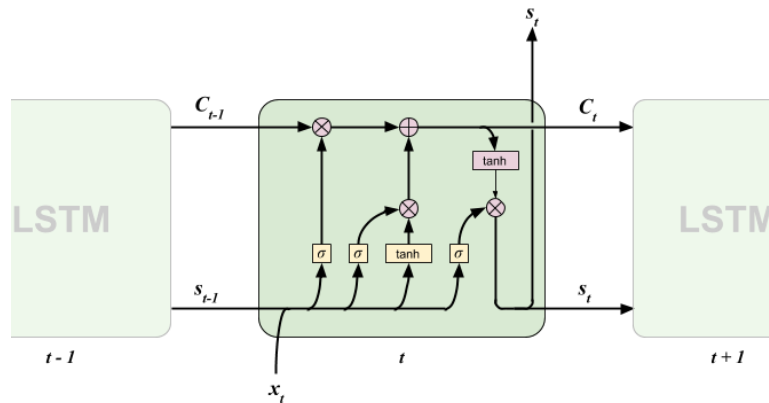


Gambar 2.2. Hidden layer pada RNN

$$st = \tanh(U \cdot xt + W \cdot st - 1) \dots\dots\dots(2.1)$$

$$\hat{y}t = \text{softmax}(V \cdot st) \dots\dots\dots(2.2)$$

Berbeda pada pemrosesan modul *LSTM* memiliki komputasi yang banyak.



Gambar 2.3 Memori pada LSTM

Dari modul *Recurrent neural network (RNN)* dan *Long Short Term Memory (LSTM)* memiliki tiga tahapan, yaitu Input nilai data pelatihan sehingga diperoleh nilai output (tahap *Feedforward*), Propagasi balik dari nilai *error* yang diperoleh (tahap *Backpropagation*), didalam kedua modul arsitekturnya terdiri dari tiga layer, yaitu:

- **Input Layer**

Berisi node-node yang mempunyai sebuah nilai masukan yang tidak berubah pada fase latih dan hanya bisa berubah jika diberikan nilai masukan baru. Node pada layer ini tergantung pada banyaknya input dari suatu pola.

- **Hidden Layer**

Layer ini tidak pernah muncul sehingga dinamakan *hidden layer*. Namun semua proses pada fase pelatihan dan fase pengenalan dijalankan di lapisan ini. Jumlah lapisan ini tergantung dari arsitektur yang akan dirancang, tetapi pada umumnya terdiri dari satu lapisan hidden layer yang saling berurutan. Didalam hidden layer terdapat perbedaan antara modul *Recurrent Neural Network (RNN)* dan *Long Short Term Memory (LSTM)* perbedaannya yaitu :

1. **Hidden layer Recurrent Neural Network (RNN) :**

Hidden layer RNN terdiri dari sel memori, satu sel memori memiliki satu gerbang dan output dengan fungsi *softmax*, satu gerbang tersebut digunakan untuk menyimpan data dari sel memori sebelumnya (s_{t-1}). untuk diproses dengan fungsi tanh bersamaan dengan data masukan sekarang (x_t). Arsitektur RNN dapat dilihat diatas pada Gambar 2 dan rumus pada persamaan (2.1) dan (2.2).

2. *Hidden Long Short Term Memory (LSTM)* terdiri dari sel memori, satu sel memori memiliki tiga *gate* yaitu :

a. ***Forget gate***

Forget gate adalah lapisan melupakan yang mengambil *output* pada waktu (h^{t-1}) dan *input* pada waktu (x^t) parameter tersebut digabungkan dan di proses dengan fungsi sigmoid. Dari output ini menghasilkan antara dua bilangan 0 atau 1 dimana $f_t = 0$ maka keadaan (state) sebelumnya akan dilupakan, sementara jika $f_t = 1$ state sebelumnya tidak berubah.

$$z_t^{f_t} = W^{f_t} \cdot \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix} \dots\dots\dots (2.3)$$

$$f_t = \sigma(z_t^f + b_f) \dots\dots\dots (2.4)$$

b. ***Input gate***

Input gate melakukan pembuatan vector baru dan disimpan di *memory cell* terdapat dua proses yaitu :

1. Proses input asli dari perhitungan *output* sebelumnya waktu (h^{t-1}) dan *input* baru waktu (x^t), diproses dengan fungsi sigmoid. Di *paper* disebut (lapisan sigmoid input) [7].

$$z_t^i = W^i \cdot \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix} \dots\dots\dots (2.5)$$

$$i_t = \sigma(z_t^i + b_i) \dots\dots\dots (2.6)$$

2. Proses memperhitungkan *output* sebelumnya dan *input* baru (x^t) diproses dengan fungsi yang berbeda yaitu tanh. Di *paper* disebut (lapisan kandidat) [7].

$$z_t^{\check{c}_t} = W^{\check{c}_t} \cdot \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix} \dots\dots\dots (2.7)$$

$$\check{c}_t = \tanh(z_t^{\check{c}_t} + b_{\check{c}_t}) \dots\dots\dots (2.8)$$

Dari proses perkalian dari kedua hasil tersebut menghasilkan vektor baru dan disimpan di *memory cell* sebagai calon masukan ($i_t \cdot \check{c}_t$), calon masukan tersebut akan dilanjutkan ke lapisan pernghubung (C_t) antara hasil f_t , calon masukan ($i_t \cdot \check{c}_t$), dan *state* pernghubung sebelumnya (C^{t-1}) diproses dengan rumus [7]. :

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t \dots\dots\dots (2.9)$$

c. **Output gate**

Output gate mengontrol seberapa banyak *state* yang lewat ke output. gerbang ini melakukan dua proses yaitu [7] :

1. Perhitungan *output* sebelumnya waktu (h^{t-1}) dan *input* baru waktu (x^t), diproses dengan fungsi sigmoid.

$$z_t^o = W^o \cdot \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix} \dots\dots\dots (2.10)$$

$$o_t = \sigma(z_t^o + b_i) \dots\dots\dots (2.11)$$

2. Proses mengontrolnya dua *state* dari *state* penghubung c_t , dan *state* output o_t diproses menghasilkan *block output* (h^t) dengan fungsi $\tanh c_t$ [7] :

$$h_t = o_t \cdot \tanh(c_t) \dots\dots\dots (2.12)$$

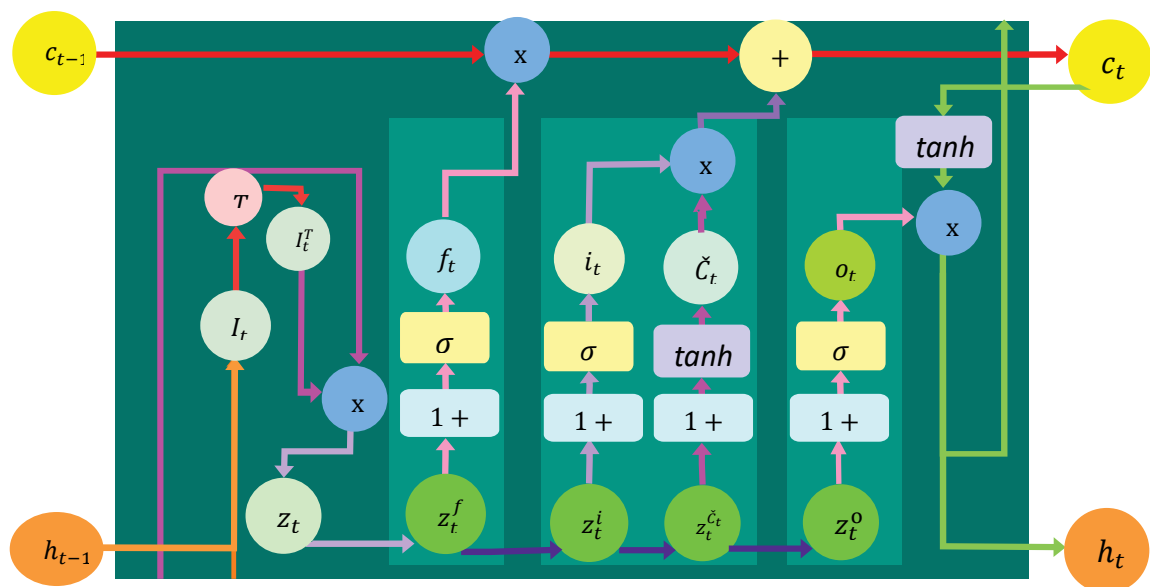
- **Output Layer**

Output layer berfungsi untuk menampilkan hasil perhitungan sistem dengan fungsi gerbang – gerbang pada *hidden layer* berdasarkan input yang diterima.

Secara umum dari modul *Recurrent neural network (RNN)* dan modul *Long Short Term Memory (LSTM)* memiliki arsitektur yang sama yaitu memiliki tiga tahapan yaitu *input*, kemudian *hidden layer* yang berproses berulang didalamnya dan akhirnya akan dilanjutkan pada tahap *output layer*. Setiap literasi dalam *Recurrent Neural Network (RNN)* maupun *Long Short Term Memory (LSTM)* dilakukan dengan dua arah yaitu *Forward* (tahap maju) dan *Backward* (tahap mundur).

2.3 Long Short Term Memory (LSTM)

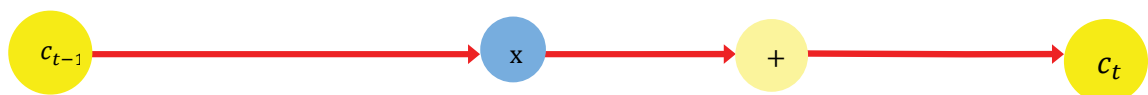
Long Short Term Memory Networks (LSTM) merupakan salah satu metode dari *Recurrent Neural Network* (RNN). LSTM diajukan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997. LSTM di *design* untuk menghindari masalah *long term dependency* yang ada pada RNN (Grave, 2014) [4]. Didalam *hidden layer* LSTM yang saling berhubungan dan saling berkaitan, terdapat proses dalam menyeleksi suatu informasi untuk diingat dalam jangka waktu pendek maupun dalam jangka waktu panjang, setiap proses dalam satu urutan akan dijelaskan sebagai berikut :



Gambar 2.4 Arsitektur LSTM

1. Status sel

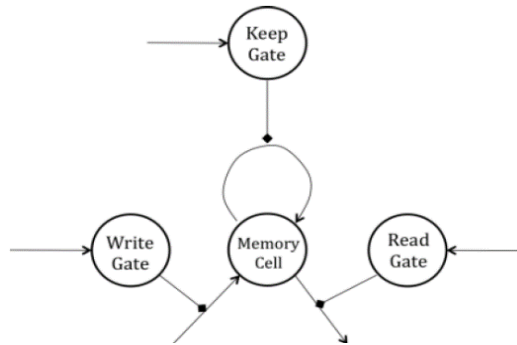
Status sel ini berjalan lurus dan menghubungkan seluruh urutan. (*Long memory*), Sel ini akan menghubungkan dari semua urutan, diawali dari urutan yang pertama hingga urutan yang berlangsung sekarang atau berproses sekarang.



Gambar 2.5 Status sel LSTM

2. Gerbang

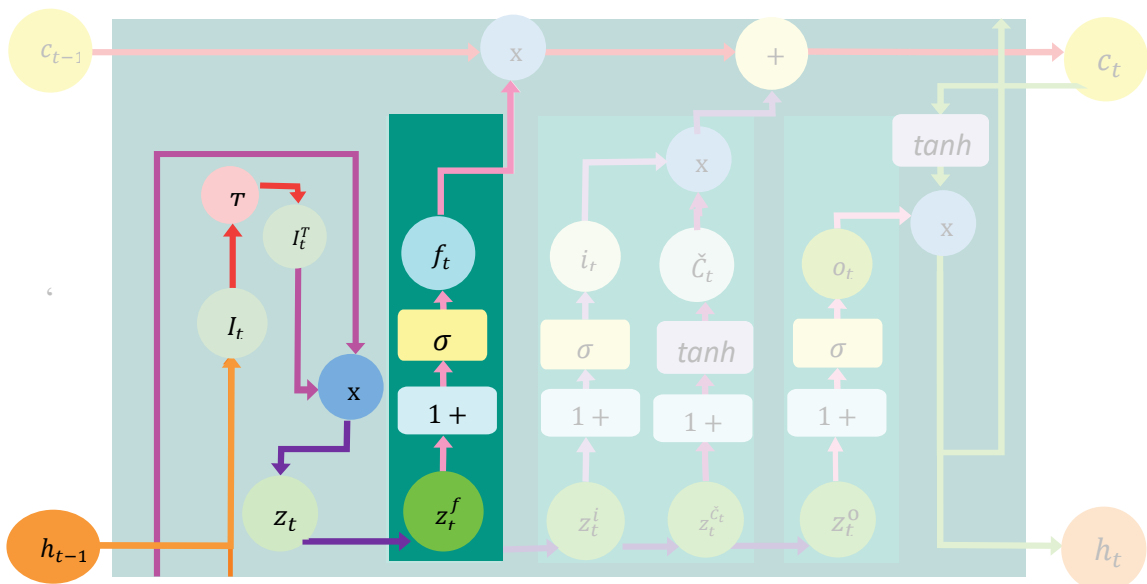
Gerbang ini bertujuan mendapatkan informasi secara opsional. Dari satu urutan terdapat 3 gerbang yaitu *Forget gate layer (keep gate)* bertujuan untuk menjaga informasi sebelumnya agar tidak sembarangan masuk kedalam keadaan urutan sekarang, *Forget gate layer (write gate)* bertujuan untuk menuliskan informasi apa yang didapat dari urutan sebelumnya dan disimpan di *memory cell* dan *read gate* bertujuan untuk menerima informasi yang sudah terseleksi sebelumnya dengan melewati gerbang – gerbang sebelumnya.



Gambar 2.6 Gerbang *LSTM*

3. Penjagaan gerbang (*Keep gate*)

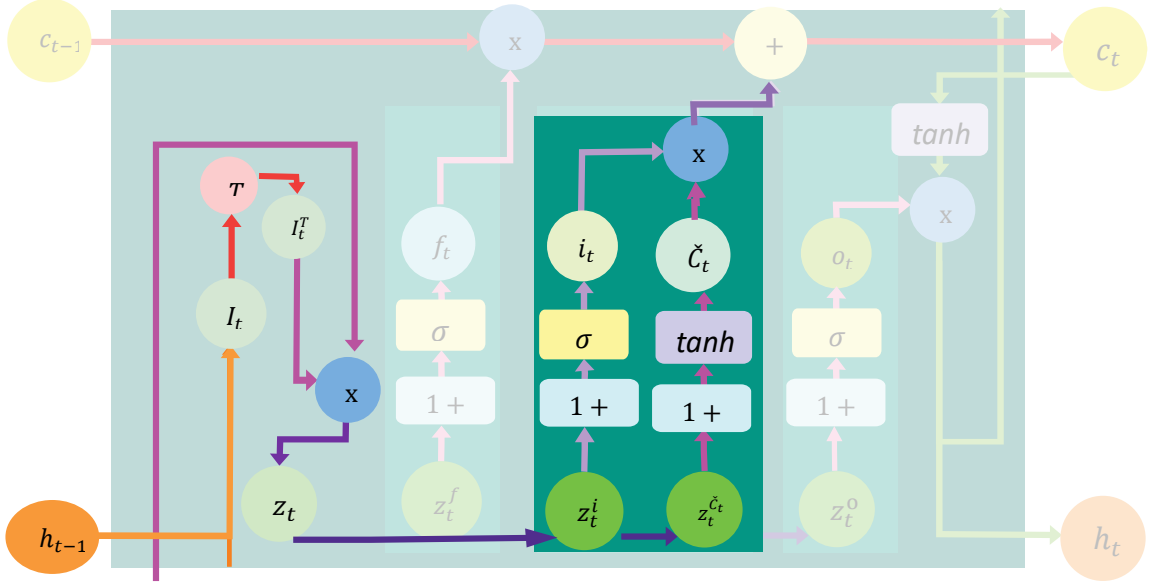
Penjagaan informasi yang akan dibuang dari urutan sebelumnya. Dari gerbang ini diberikan *output* antara dua bilangan 0 atau 1 dimana $f_t = 0$ maka keadaan sebelumnya akan dilupakan, sementara jika $f_t = 1$ keadaan sebelumnya tidak berubah. Dijabarkan dengan persamaan (2.4).



Gambar 2.7 Penjagaan gerbang (*Keep gate*) f_t *LSTM*

4. Menambahkan informasi baru (*write gate*)

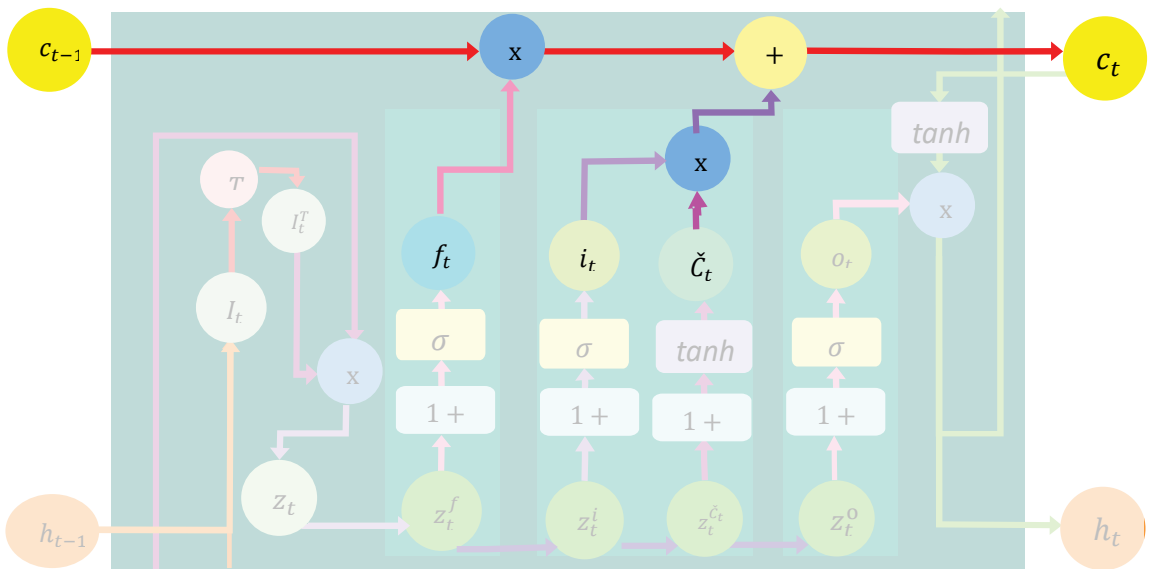
Menambahkan informasi baru Penentuan apa yang disimpan dalam keadaan sel, di dalam gerbang ini terdapat 2 lapisan yaitu, lapisan gerbang input untuk menambahkan nilai baru dari input, dijabarkan dengan persamaan (2.6), dan lapisan tanh untuk membuat vektor nilai kandidat baru, dijabarkan dengan persamaan (2.8).



Gambar 2.8 Menambahkan informasi baru (*write gate*) LSTM

5. Update cell state (c_t)

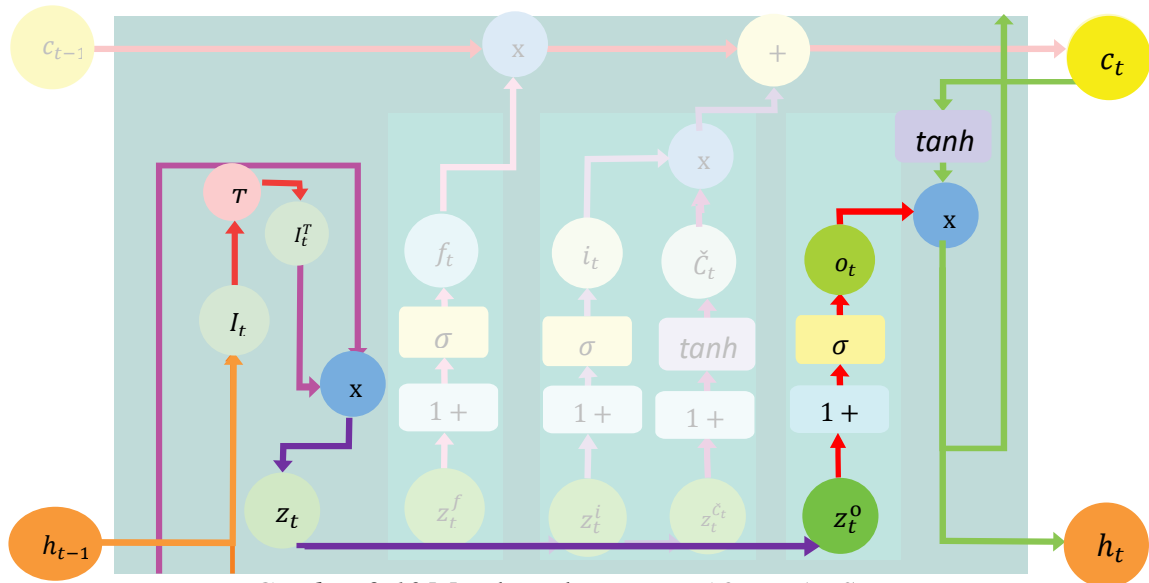
Proses melupakan hal-hal yang sudah diseleksi untuk dilupakan sebelumnya $f_t * \check{c}_{t-1}$ dan proses menambahkan informasi yang sudah ditentukan sebelumnya $i_t * \check{c}_t$ Secara rumus dapat dituliskan seperti persamaan (2.9)



Gambar 2.9 Update cell state LSTM

6. Membaca keputusan (*Output*)

Proses ini akan menentukan bagaimana sel mana yang akan dihasilkan dengan bantuan lapisan gerbang *output* dan lapisan *tanh* akan memberikan nilai antara -1 dan +1



Gambar 2.10 Membaca keputusan (*Output*) LSTM

Lapisan gerbang output dijabarkan dengan persamaan (2.11)

Lapisan \tanh dijabarkan dengan persamaan (2.12) (*Short memory*)

2.3.1 Algoritma pelatihan *Long Short Term Memory Networks*

Algoritma pelatihan *Long Short Term Memory Networks* pada dasarnya terdiri dari beberapa tahapan yaitu :

- Input* nilai data *training* panjang urutan (*Sequence length*)
- Forward LSTM* (tahap maju) sampai diperoleh nilai *output*
- Backpropagation through time LSTM* (tahap mundur) sampai diperoleh nilai error yang dipeoleh.
- Penyesuaian bobot (*gradient descent*) sampai mendapatkan bobot minimum

Dari tahapan diatas diulangi terus- menerus sampai mendapatkan nilai *error* yang diinginkan.

Berikut langkah-langkah pelatihannya adalah :

Langkah 0 :

- Persiapkan data normalisasi
- Inisialisasi dimensi *input* dan *output* (x_t, y_t)
- Tetapkan *Epcchs* (satu presentasi lengkap dari proses *forward* sampai *Backward* untuk pemrosesan data yang akan dipelajari ke mesin pembelajaran)
- Inisialisasi bobot (bobot awal dengan nilai random yang cukup kecil)
- Tetapkan maksimum target *error* dan *learning rate*
- Tetapkan panjang urutan (*sequence length*)

Langkah 1 :

- Penggabungan antara input data dan block output dari langkah sebelumnya:

$$I^t = \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix} \dots\dots\dots (2.13)$$

I_t : Hasil penggabungan *input*

h^{t-1} : Hasil block output sebelumnya

x_t : Input data yang akan diolah

- Block input dengan mempersiapkan input $f_t, i_t, \check{c}_t, o_t$:

$$W_t = \begin{bmatrix} w^f & U^f \\ w^i & U^i \\ w^c & U^c \\ w^o & U^o \end{bmatrix} \dots\dots\dots (2.14)$$

Catatan :

- Baris W_t jika simpul 1 maka $4 \times 1 = 4$
- Kolom W_t jika *size* input 2 maka $2 + \text{simpul} = 3$
- Jadi ordo matrik W_t adalah (4,3)

$$z_t = \left(W_t \cdot \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix} \right) \dots\dots\dots (2.15)$$

$$z_t = \begin{bmatrix} w^f & U^f \\ w^i & U^i \\ w^c & U^c \\ w^o & U^o \end{bmatrix} \cdot \begin{bmatrix} b \\ x^t \\ h^{t-1} \end{bmatrix} \dots\dots\dots (2.16)$$

$$z_t = \begin{bmatrix} z_t^f \\ z_t^i \\ z_t^{\check{c}} \\ z_t^o \end{bmatrix} \dots\dots\dots(2.17)$$

Keterangan :

- z_t : Hasil *input (hidden layer)* terdapat 4 baris untuk *input*
- z_t^f : *input (hidden layer)* pada gerbang *forget* seperti persamaan (2.3)
- z_t^i : *input (hidden layer)* pada gerbang *input* seperti persamaan (2.5)
- $z_t^{\check{c}}$: *input (hidden layer)* pada *candidate state* seperti persamaan (2.7)
- z_t^o : *input (hidden layer)* pada gerbang *output* seperti persamaan (2.10)
- W_t : Bobot (weight) yang menghubungkan input layer
- U : Data informasi

2.3.2 Feedforward

Langkah 2 :

- a. Gerbang lupa (*forget gate*):

$$f_t = \sigma(z_t^f + b_f) \dots\dots\dots(2.3)$$

Keterangan :

- f_t : Hasil perhitungan gerbang lupa (*forget gate*)
- z_t^f : *Block Input forget gate*
- z_i : Nilai *input* dari *hidden output*
- b_f : Bias
- σ : Logistic sigmoid ($f(x) = \frac{1}{1+e^x}$)

- b. Gerbang pembuatan vektor baru dan disimpan di *memory cell*:

$$3. i_t = \sigma(z_t^i + b_i) \dots\dots\dots(2.6)$$

$$4. \check{c}_t = \tanh(z_t^{\check{c}} + b_{\check{c}}) \dots\dots\dots (2.8)$$

Keterangan :

- i_t : Hasil perhitungan gerbang *input (input gate)*
- \check{c}_t : Proses tanh (*candidate gate*)
- tanh* : Fungsi tanh digunakan membuat vektor baru

- c. Keadaan penghubung (*state cell*) :

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t \dots\dots\dots (2.9)$$

Keterangan :

c_t : Hasil perhitungan keadaan penghubung (*connecting state*)

c_{t-1} : Keadaan penghubung sebelumnya

- d. Gerbang *output* (*output gate*) :

$$o_t = \sigma(z_t^o + b_o) \dots\dots\dots (2.11)$$

Keterangan :

o_t : Hasil perhitungan gerbang *output* (*output gate*)

- e. *Block output*

$$h_t = o_t \cdot \tanh(c_t) \dots\dots\dots (2.12)$$

Keterangan :

h_t : Hasil perhitungan *output* dan hasil prediksi

2.3.3 Backpropagation Through Time (BPTT)

Langkah 3 :

$$E = (y_t - \hat{y}_t)$$

Keterangan :

y_t : Harga real (data asli)

\hat{y}_t : Harga prediksi (data prediksi)

Pada proses forward diberikan $h^t = o^t \odot \tanh(c^t)$ dan ditemukan $\delta o^t, \delta c^t$

- a. Hitung informasi *derivative* gerbang output δo^t :

$$\delta o^t = \delta h^t \odot \tanh(c^t) \dots\dots\dots (2.18)$$

Keterangan :

δo^t : Menghitung informasi *derivative* gerbang output

δh^t : Informasi *derivative* output block

c^t : keadaan penghubung (*connecting state*)

- b. Hitung informasi *derivative* keadaan penghubung (*cell state*) δc^t :

$$\delta c^t = \delta h^t \odot o^t \odot (1 - \tanh^2(c^t)) \dots\dots\dots(2.19)$$

Keterangan :

δc^t : Menghitung informasi *derivative* keadaan penghubung

o^t : Gerbang *Output*

Langkah 4 :

Pada proses forward diberikan $c^t = i^t \odot a^t + f^t \odot c^{t-1}$ dan ditemukan $\delta i^t, \delta a^t, \delta f^t, \delta c^{t-1}$

- a. Hitung informasi *derivative* gerbang input (*input gate*) δi^t :

$$\delta i^t = \delta c^t \odot a^t \dots\dots\dots(2.20)$$

Keterangan :

δi^t : Menghitung informasi *derivative input gate*

δc^t : Informasi *derivative* keadaan penghubung

a^t : *candidate gate* (\check{C}_t)

- b. Hitung informasi *derivative* gerbang *candidate connecting state* a^t

$$\delta a^t = \delta c^t \odot i^t \dots\dots\dots(2.21)$$

Keterangan :

δa^t : Informasi *derivative candidate connecting state*

δc^t : Informasi *derivative* Keadaan penghubung

i^t : *input gate*

- c. Hitung informasi *derivative* gerbang *forget gate* δf^t

$$\delta f^t = \delta c^t \odot c^{t-1} \dots\dots\dots(2.22)$$

Keterangan :

δf^t : Informasi *derivative forget gate*

δc^t : Informasi *derivative* keadaan penghubung

c^{t-1} : Keadaan penghubung sebelumnya

- d. Hitung informasi derivative Keadaan penghubung sebelumnya

$$\delta c^{t-1} = \delta c^t \odot f^t \dots\dots\dots (2.23)$$

Keterangan :

δc^{t-1} : Informasi *derivative* keadaan penghubung sebelumnya

δc^t : Informasi *derivative* keadaan penghubung

f^t : Gerbang *forget gate*

Langkah 5 :

Pada proses forward diberikan Forward Pass: $z^t = \begin{bmatrix} \hat{a}^t \\ \hat{i}^t \\ \hat{f}^t \\ \hat{o}^t \end{bmatrix} = W \times I^t$ maka pada fase

backward didapat $\delta z^t = \delta a^t, \delta i^t, \delta f^t, \delta o^t$.

- a. Hitung *derivative block input (candidate gate)* :

$$a^t = \check{C}_t$$

$$\delta z \hat{a}^t = \delta a^t \odot (1 - \tanh^2(\hat{a}^t)) \dots\dots\dots (2.24)$$

Keterangan :

$\delta z \hat{a}^t$: Informasi *derivative input block candidate gate*

δa^t : Informasi *derivative candidate gate*

\hat{a}^t : *block input candidate gate*

- b. Hitung *derivative block input (input gate)*:

$$\delta z \hat{i}^t = \delta i^t \odot I^t \odot (1 - i^t) \dots\dots\dots (2.25)$$

Keterangan :

$\delta z \hat{i}^t$: Informasi *derivative input block input gate*

δi^t : Informasi *derivative input gate*

i^t : *input gate*

- c. Hitung *derivative block input (forget gate)* :

$$\delta z \hat{f}^t = \delta f^t \odot f^t \odot (1 - f^t) \dots\dots\dots (2.26)$$

Keterangan :

$\delta z \hat{f}^t$: Informasi *derivative input block forget gate*

δf^t : Informasi *derivative forget gate*

f^t : *forget gate*

- d. Hitung *derivative block input (output gate)* :

$$\delta z \hat{o}^t = \delta o^t \odot o^t \odot (1 - o^t) \dots\dots\dots (2.27)$$

$\delta z \hat{o}^t$: Informasi *derivative input block output gate*

δo^t : Informasi *derivative output gate*

o^t : *output gate*

- e. Penggabungan dari $\delta z \hat{a}^t, \delta z \tilde{t}^t, \delta z \hat{f}^t, \delta z \hat{o}^t$ menjadi satu untuk menghasilkan δz^t dari *forward* terdapat Forward Pass: $z^t = W \times I^t$ maka perlu pengembalian z^t ke δz^t :

Dari forward didapat persamaan (2.17) maka fase backward didapat persamaan

$$\delta z^{tT} = [\delta z \hat{a}^t, \delta z \tilde{t}^t, \delta z \hat{f}^t, \delta z \hat{o}^t]^T \dots\dots\dots (2.28)$$

$$\delta z^{tT} = \begin{bmatrix} \delta z \hat{a}^t \\ \delta z \tilde{t}^t \\ \delta z \hat{f}^t \\ \delta z \hat{o}^t \end{bmatrix} \dots\dots\dots (2.29)$$

2.3.4 Mencari Bobot dari proses backpropagation urutan (sequence)

Langkah 6 :

Dari tahap *forward pass* didapat persamaan (2.15) dan diberikan δz^t dari langkah sebelumnya maka dicari δW^t dan δh^{t-1}

- a. Persamaan (2.29) didapat untuk menghitung δI^t

$$\delta I^t = \delta z^t \cdot W^{tT} \dots\dots\dots (2.30)$$

$$\delta I^t = \begin{bmatrix} \delta z \hat{a}^t \\ \delta z \tilde{t}^t \\ \delta z \hat{f}^t \\ \delta z \hat{o}^t \end{bmatrix} \cdot [W^c \ W^i \ W^f \ W^o] \dots\dots\dots (2.31)$$

- b. Dari persamaan (2.31), didapat untuk menghitung *derivative block output* (δh^{t-1})

$$\delta h^{t-1} = [\delta W^o \cdot \delta z \hat{o}^t] \dots\dots\dots (2.32)$$

- c. Mencari δW^t *derivative* pada bobot yang akan menjadi model dalam prediksi selanjutnya

$$\delta W^t = \delta z^t \cdot I^{tT} \dots\dots\dots (2.33)$$

$$\delta W^t = \begin{bmatrix} W^c \cdot \delta z \hat{a}^t \\ W^i \cdot \delta z \hat{i}^t \\ W^f \cdot \delta z \hat{f}^t \\ W^o \cdot \delta z \hat{o}^t \end{bmatrix} \cdot [W^c \cdot \delta z \hat{a}^t, W^i \cdot \delta z \hat{i}^t, W^f \cdot \delta z \hat{f}^t, W^o \cdot \delta z \hat{o}^t] \dots (2.34)$$

d. 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]

$$\delta W^t = \begin{bmatrix} W^c \cdot \delta z \hat{a}^t \\ W^i \cdot \delta z \hat{i}^t \\ W^f \cdot \delta z \hat{f}^t \\ W^o \cdot \delta z \hat{o}^t \end{bmatrix} \cdot [W^c \cdot \delta z \hat{a}^t, W^i \cdot \delta z \hat{i}^t, W^f \cdot \delta z \hat{f}^t, W^o \cdot \delta z \hat{o}^t] (2.34)$$

$$\frac{\partial E^t}{\partial W} = \delta W^t \dots (2.35)$$

$$\delta W^t(\text{baru}) = \delta W^t(\text{lama}) + \delta W^{t+1} \dots (2.36)$$

Keterangan :

δW^t = derivative bobot

$\delta W^t(\text{baru})$ dihitung sampai t =1

2.3.5 Optimasi untuk update bobot Gradient Descent

Langkah 7 : *Gradient descent weight optimization*

Menghitung *Gradient descent* bobot optimasi dari parameter, bobot dan bias, dalam jaringan saraf diperbarui menggunakan data pelatihan sehingga cost average dari semua contoh pelatihan diminimalkan [9].

Dapat dituliskan dengan persamaan :

$$W(\text{final}) = W(\text{Lama}) - \text{learning_rate} * \delta W^1 \dots (2.37)$$

Keterangan :

learning_rate : Parameter untuk mengontrol perubahan bobot selama pelatihan. Semakin besar learning rate, maka jaringan syaraf tiruan akan semakin cepat belajar tetapi hasilnya kurang akurat. Semakin kecil learning rate, maka jaringan syaraf tiruan akan semakin lambat belajar tetapi hasilnya lebih akurat [10]

$W(\text{final})$ = bobot ini akan digunakan pemodelan.

2.3.6 Menghitung *error* dari perbandingan data asli dengan data prediksi

Langkah 8 : Hitung total *error*

Menghitung total *error* menggunakan *mean squared error* dengan rumus perhitungan:

$$MSE = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \dots\dots\dots (2.38)$$

Keterangan :

MSE : Merupakan total *error*

y_i : Merupakan target *output* ke-*i*

ŷ_i : Merupakan *output* dari pelatihan ke-*i*

Fase tersebut diulang hingga kondisi *error* terpenuhi.

2.4 Penelitian Terkait

Table 1. Tabel Penelitian Terkait

Peneliti, Tahun	Permasalahan	Metode	Hasil Penelitian	Perbedaan dengan penelitian yang akan dilakukan
1. Johanes Lusikooy, dkk. (2020) [11]	Memprediksi harga tutup saham PT. Garuda Indonesia,Tbk. 25 Oktober 2016 sampai 7 November 2016	ARIMA	Hasil nilai MSE terkecil yaitu 63,7	Paper : 1. Saham yang diteliti yaitu PT. Garuda Indonesi,Tbk, dengan metode ARIMA. Tugas Akhir : 1. Saham yang diteliti yaitu PT Semen Indonesia (persero) Tbk, dengan metode LSTM
2. Eko Riyanto. (2017). [12]	Prediksi harga saham selama lima tahun, mulai dari tahun 2012 sampai dengan tahun 2016..	Backpropagation.	Hasil peramalan yang akurat lebih dari 98%	Paper : 1. Saham yang diteliti yaitu Astra Graphia, Astra Internasional, Indofood Sukses Makmur, dan Telekomunikasi Indonesia, dengan metode backpropagation. Tugas Akhir : 1. Saham yang diteliti hanya PT Semen Indonesia (persero) Tbk, dengan metode LSTM
3. Yudi Ramdhani, Ade Mubarok (2019). [13]	Prediksi harga saham Antm.Jk penutupan pada tahun 2013	SVM Model Regresi	Hasil nilai terendah RMSE dengan nilai 22.662	Paper : 1. Saham yang diteliti yaitu PT Aneka Tambang Tbk, dengan metode SVM Model Regresi . Tugas Akhir : 1. Saham yang diteliti hanya PT Semen Indonesia (persero) Tbk, dengan metode LSTM

Peneliti, Tahun	Permasalahan	Metode	Hasil Penelitian	Perbedaan dengan penelitian yang akan dilakukan
4. Ahmad Fauzi. (2019). [14]	Memprediksi Index harga saham Jakarta Islamic Index (JII)	LSTM	Epoch 20, MSE sebesar 0.00019 dan RMSE sebesar 0.014	Paper : 1. Objek yang di teliti yaitu Index harga saham Jakarta Islamic Index (JII). Tugas Akhir : 1. Objek saham harga saham Indonesia PT Semen Indonesia (persero) Tbk.
5. Divit Karmiani, dkk. (2019). [3]	Melakukan pembuatan momentum pada harga saham, penggabungan harga saham yang masuk disektor technology, dan juga index NASDAQ.	Backpropagation, LSTM, dan SVM	- Mean Accuracy SVM 67.121 % - Mean Accuracy LSTM 69.04 % - Mean Accuracy Backpropagation 67.43 %	Paper : 1. Objeck yang diteliti sektor technology, dan juga index NASDAQ Tugas Akhir : 1. Objeck yang diteliti saham PT Semen Indonesia (persero) Tbk

Berdasarkan penelitian diatas yang berjudul *Comparison of Predictive Algorithms: Backpropagation, SVM, LSTM and Kalman Filter for Stock Market* menyimpulkan bahwa prediksi saham menggunakan LSTM lebih baik dari backpropagation, dan SVM, penelitian yang berjudul *Forecasting saham syariah dengan menggunakan LSTM* sama-sama menyimpulkan juga bahwa prediksi saham India dan saham Indonesia dengan menggunakan *LSTM* menghasilkan hasil yang baik.

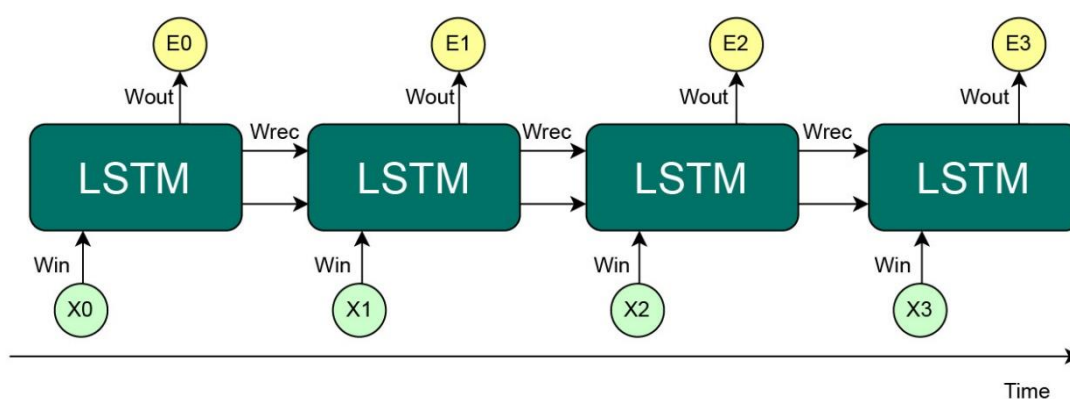
BAB III METODE USULAN

3.1 Desain Rancangan

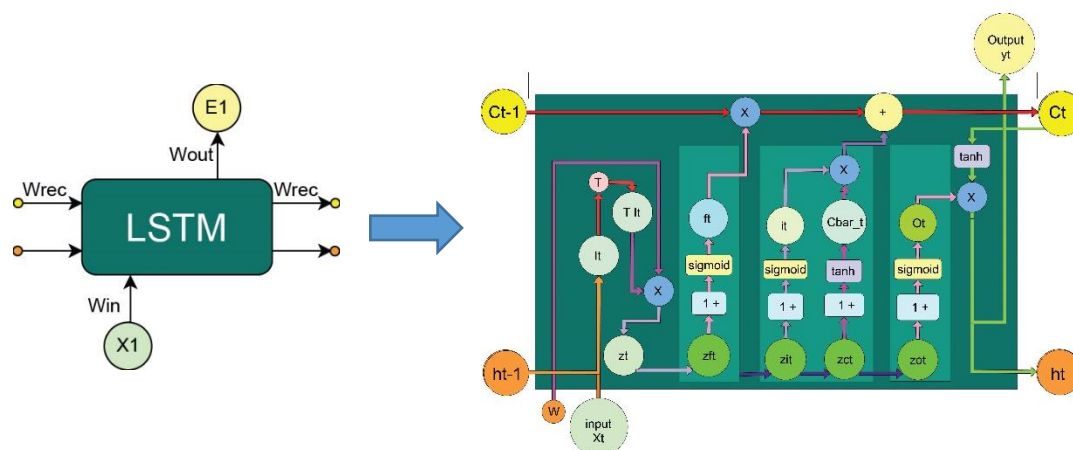
Desain rancangan sistem dibuat agar sistem yang dibangun akan lebih mudah untuk menganalisa jika terjadi kesalahan ataupun ketidaksesuaian dengan kondisi yang diharapkan.

3.1.1 Rancangan Arsitektur LSTM

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



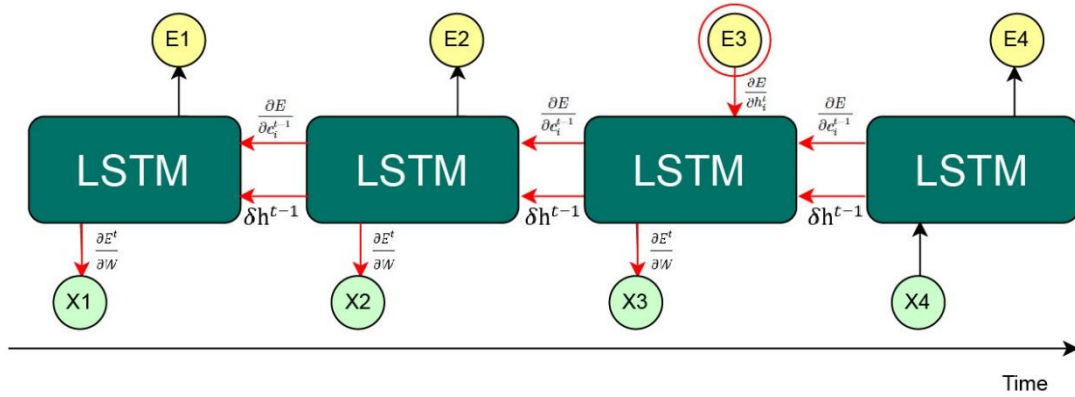
Gambar 3.11 Arsitektur *LSTM forward*



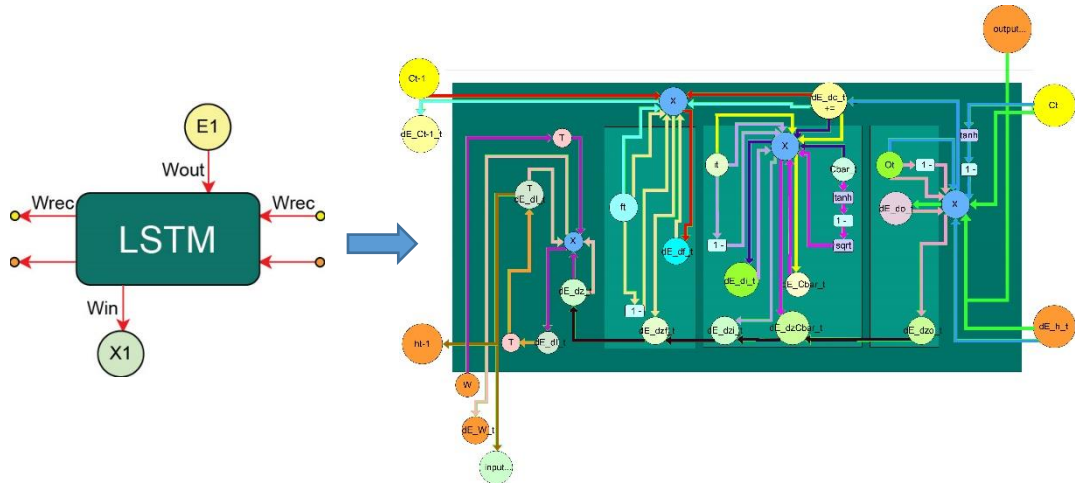
Gambar 3.12 Penjabaran dari *Hidden Layer LSTM*

Pada gambar 5 setiap urutan x_0, x_1, \dots, x_t 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 ditunjukkan pada gambar 3.1 didapatlah bobot baru (W^o), bobot tersebut memiliki hasil *error* yang besar, jadi dibutuhkanlah proses *Backward* untuk mendapatkan *error* bobot terbaik.



Gambar 3.13 Arsitektur *LSTM Backward*



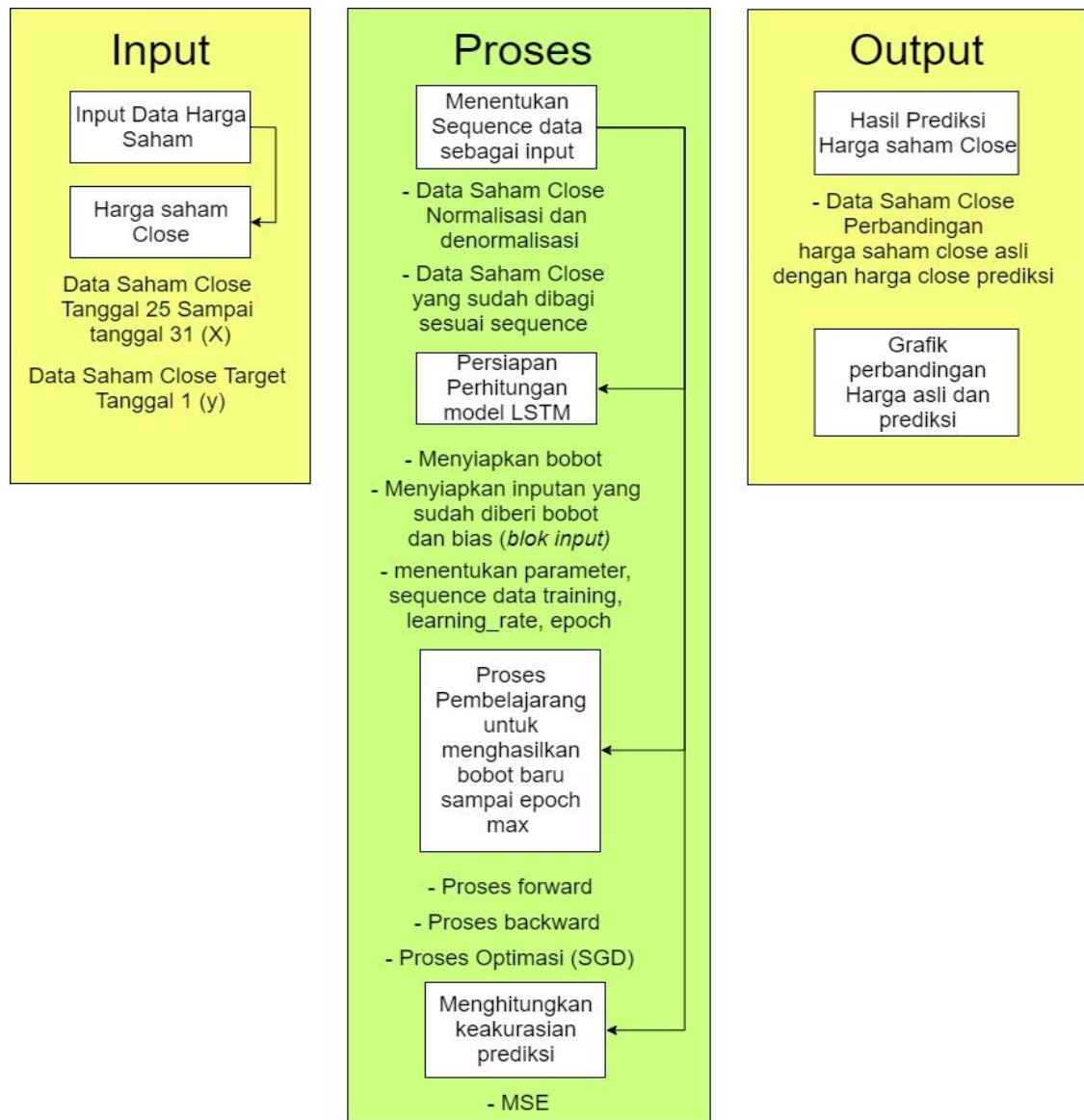
Gambar 3.14 Penjabaran dari *Hidden Layer LSTM Backward*

Pada gambar 3.3 ditunjukkan 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* (δh^{t-1}), *derivative* keadaan penghubung (*cell state*) δc^{t-1} , dan bobot *derivative* (δW^t).

Derivative block input (δh^{t-1}) dan *derivative* keadaan penghubung (*cell state*) δc^{t-1} , ini akan menghubungkan dari urutan ke urutan sebelumnya sampai ke urutan pertama untuk mendapatkan *derivative minimum*, Dan tiap urutan hidden layer menghasilkan bobot *derivative* (δW^t) 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

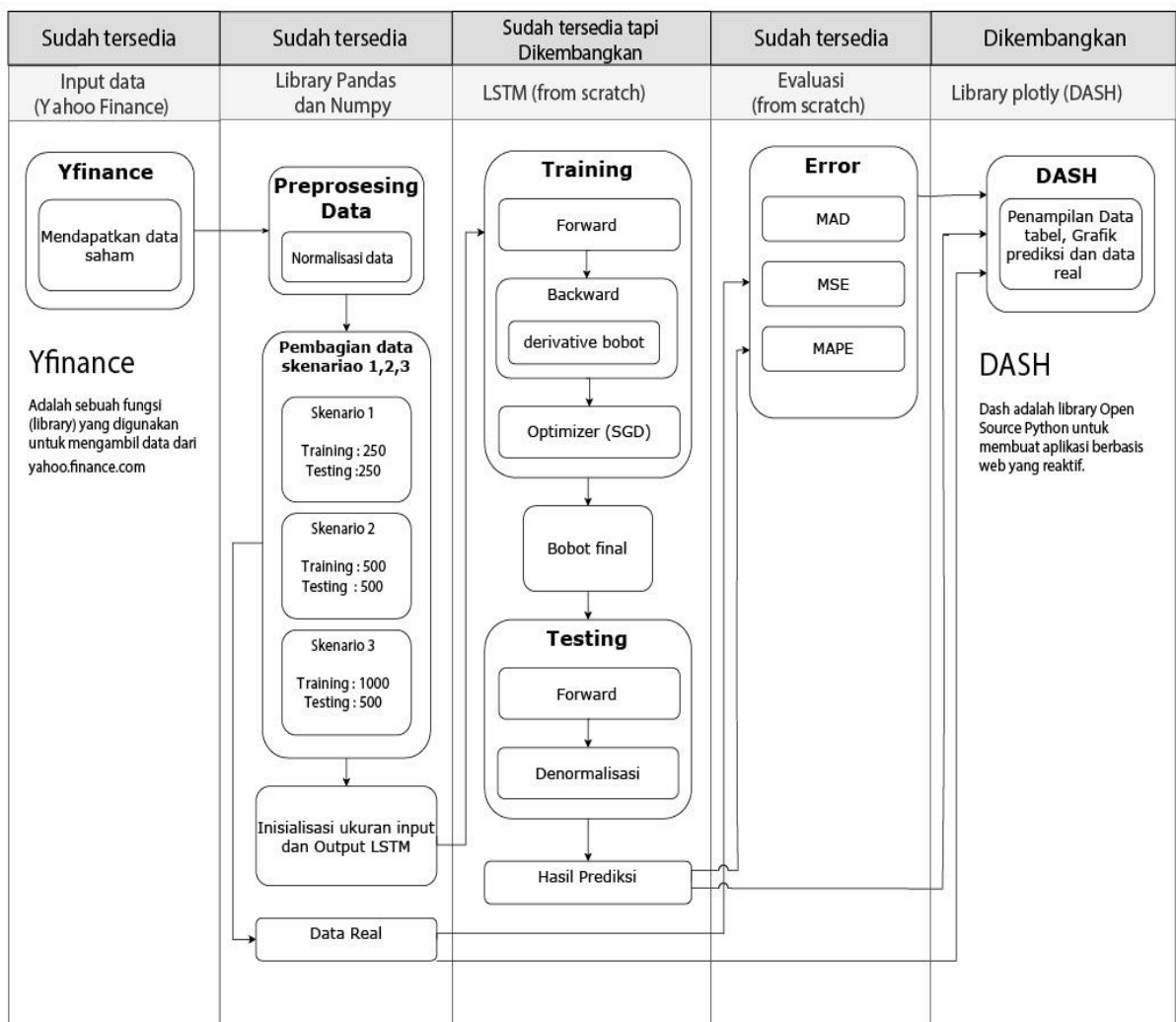
Rancangan sistem menjelaskan proses berjalannya program. Dimulai dengan pengambilan data harga saham. Kemudian diolah menjadi dataset untuk diproses menjadi data *training*.

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 digunakan 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 menggunakan 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 3.16 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*
 - a. *Library Pandas* adalah *library open source*, struktur data yang mudah digunakan dan alat analisis data untuk bahasa pemrograman Python.
 - b. *Library Numpy* adalah *library open source* untuk komputasi ilmiah dengan python yang mudah digunakan salah satunya menghitung objek array N-dimensi yang baik.

Dalam menyiapkan data untuk proses LSTM digunakan library dimana library pandas digunakan untuk membaca data dan menyeleksi data yang akan dieksekusi dalam program. Dan library Numpy digunakan untuk proses matematis seperti operasi matriks, penggabungan matriks, dan lainnya.

3. 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.

4. Evaluasi (from scratch)

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

5. GUI (Graphical user interface)

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

3.2 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 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 penelitian ini menggunakan data yang terdapat di Alphavantage 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

Penelitian dalam menerapkan metode LSTM terdapat beberapa proses yang diataranya yaitu :

- a. Pembuatan data untuk LSTM:
 - Normalisasi data
 - data deret waktu yang dinormalisasi

- b. 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

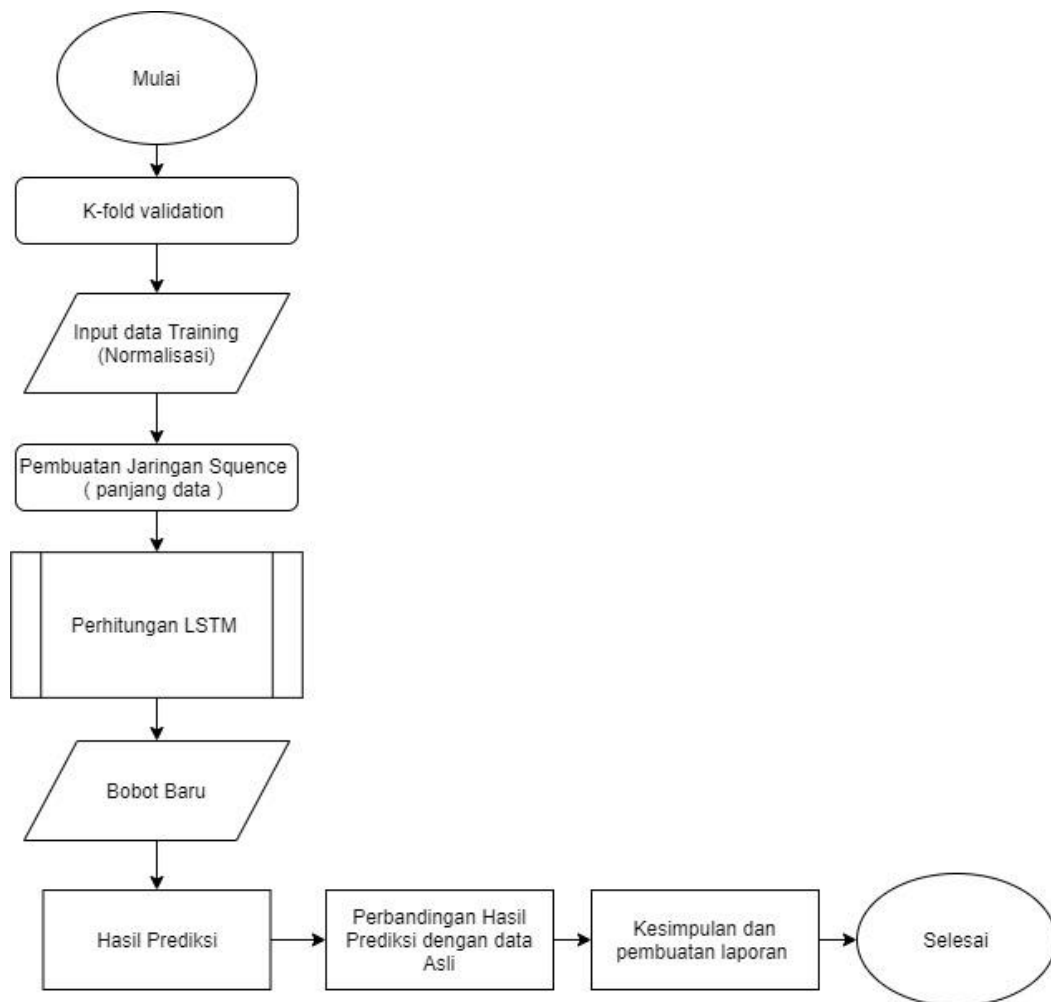
Hasil dari prediksi harga saham, saham akan ditampilkan dalam bentuk grafik perbandingan harga dari harga prediksi dengan harga aktual dalam meprediksi harga saham.

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*.
3. Menentukan panjang urutan (*Sequence 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.
4. 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.

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

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

6. 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*.

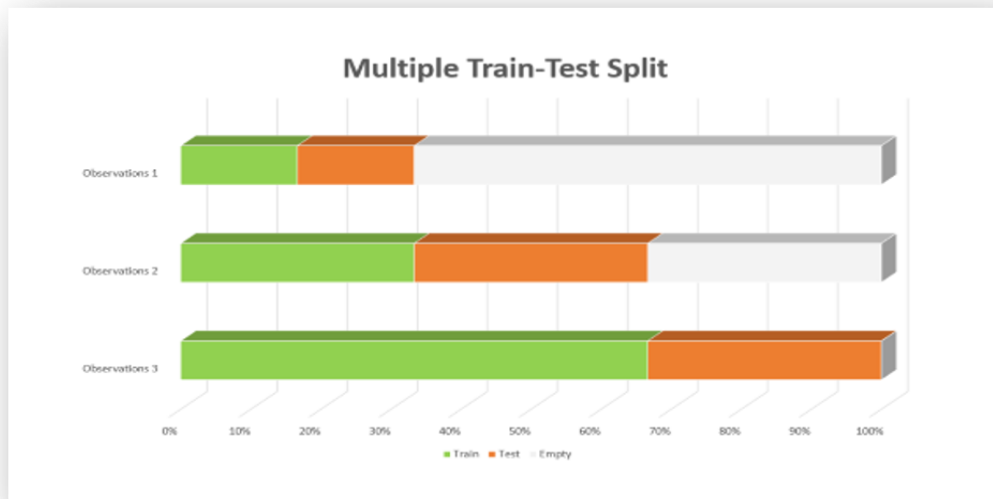
7. Menghasilkan *output*

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

8. Selesai

3.6.1 Flowchart Multiple Train-Test Split

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



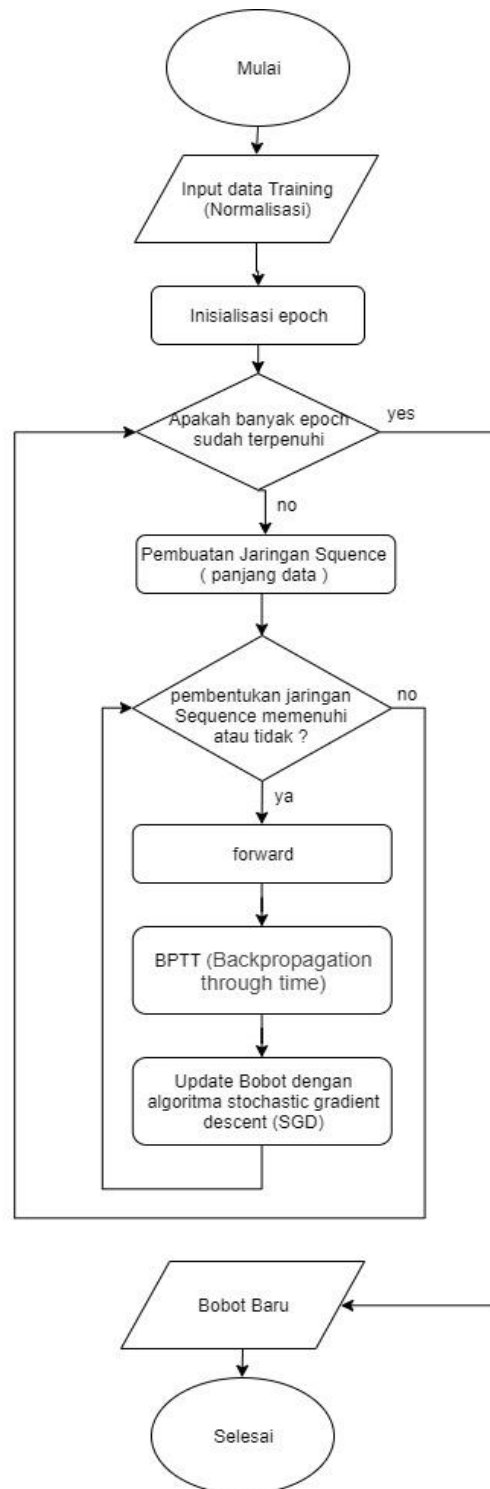
Gambar 3.18 Multiple Train-Test Split

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

Misal ditentukan pembagian 3 skenario sehingga akan didapat *dataset* skenario 1, 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.

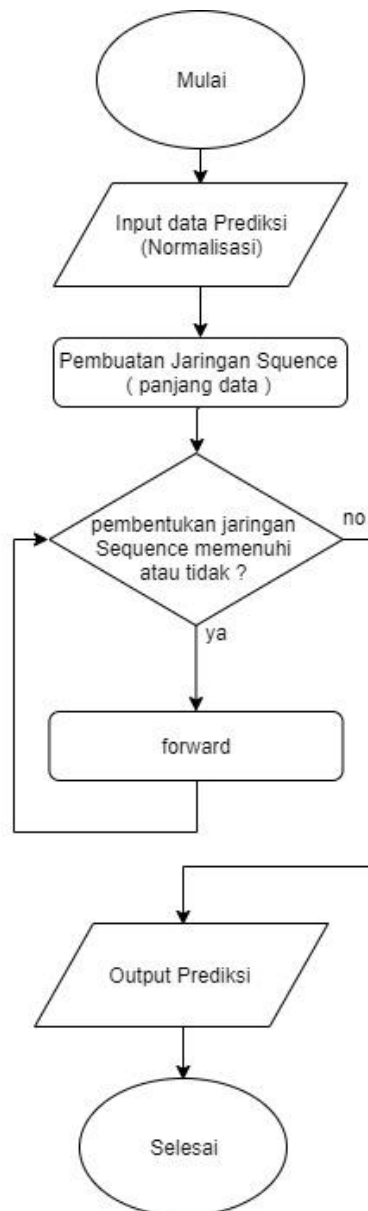


Gambar 3.19 *Flowchart* pelatihan *LSTM*

Pelatihan ini menggunakan dua looping yaitu looping pertama digunakan seberapa banyak pelatihan dalam setiap urutan (*sequence*) dan looping kedua digunakan untuk menghitung proses pembelajaran dalam pembelajaran *LSTM* menerapkan *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*

3.7 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 (*Sequence 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

$$f(x) = \frac{x_i - x_{min}}{x_{max}}$$

Tabel 5 Data *Training* dan data *Testing* Harga close Saham dinormalisasi

t	<i>Training Close</i>	<i>Testing Close</i>
1	0,0303030	0,0161616
2	0,0202020	0,0262626
3	0,0262626	0,0222222
4	0,0181818	0,0242424
5	0,0000000	?

Data untuk proses training dan testing memiliki perbedaan jumlah dikarenakan 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*.

3.7.1 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.

3.7.1.1 Fase-fase setiap epochnya

A. 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 (\hat{y}_t) tersebut dibandingkan dengan target yang harus dicapai (y_t). Jika Selisih ($y_t - \hat{y}_t$) menghasilkan kesalahan yang besar maka dilanjutkan ke fase *backward*.

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

1. Input Layer

Didalam input layer terdapat 3 masukan dalam LSTM yaitu :

a. *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*).

b. *Time steps*

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

c. *Features*

Fitur adalah kolom dalam data, dalam skenario ini hanya melibatkan 1 fitur saja yaitu harga *close* saham.

2. Inisialisasi bobot

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

Contoh data sample diurutkan pertama atau langkah pertama:

Tabel 6 Data Sample

Urutan (t)	Bias	x^t (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, didapatkan ukuran bobot matriks dengan ordo (4×3).

- bobot (W) dengan angka random :

Tabel 7 Bobot Awal

W^f	-0,245714286	0,029262045	0,184398087
W^i	0,868020398	-0,379580925	0,079506914
$W^{\check{c}}$	-0,206444161	-0,085253247	0,25112624
W^o	0,842874383	0,907722829	-0,593738792

3. Persiapan block input (z_t)

Pertama penggabungan data sample dengan block output sebelumnya (h^{t-1}) untuk mencari I^t

- Penggabungan data sample tranpose dengan *block output* (h^{t-1}), menggunakan persamaan (2.13)

Tabel 8 Penggabungan data sample tranpose dengan *block output* (h^{t-1})

t	I^1	I^2	I^3	I^4	I^5	I^6
b	-1	1	1	-1	1	1
$x^{t=1}$	-0,0303	0,0202	0,0263	-0,0182	0	0,017
h^{t-1}	0	0,0211	-0,093	-0,1678	-0,015	-0,1352

h^{t-1} sama dengan 0 dikarenakan hasil *block output* dari proses hidden layer sebelumnya (h^{t-1}) belum ada, dan untuk (h^{2-1}) adalah hasil block output dari proses hidden layer urutan ke pertama dan seterusnya.

- Perkalian antara I^t dengan bobot W^f , W^i , $W^{\check{c}}$, W^o Menggunakan persamaan (2.16)

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

z_t	z_1	z_2	z_2	z_3	z_4	z_5
z_t^f	0,209028	-0,241	-0,262	0,276	-0,248	-0,2702
z_t^i	-0,85652	0,8620	0,851	-0,85	0,867	0,8511
$z_t^{\check{c}_1}$	0,244828	-0,203	-0,232	0,250	-0,210	-0,2418
z_t^o	-0,87038	0,8487	0,922	-0,959	0,852	0,938

Setelah didapat block input (z_1) 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).

4. Hidden layer

Didalam hidden layer LSTM memiliki 3 gerbang f_t , (i_t , \check{c}_t), o_t dan dua penghubung yaitu dari semua informasi *long term* (C_t) dan *short term*

(block output h_t). Jadi dalam proses *hidden layer* ini menghitung $f_t, (i_t, \check{C}_t), o_t, C_t, h_t$.

Ditemukan dari proses *block input* (z_t) menghasilkan $z_t^f, z_t^i, z_t^{\check{C}}, z_t^o$ yang ditunjukkan pada tabel 9.

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

Langkah 1 Menghitung Gerbang lupa dengan persamaan (2.3) :

$$f_1 = \sigma(0,20903) = 0,552067456$$

f_1, f_2, \dots, f_5 melakukan proses sama sesuai dengan rumus diatas.

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

	f_1	f_2	f_3	f_4	f_5
f_t	0,55207	0,449	0,4422	0,5622	0,44765

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

- $i_1 = \sigma(-0,85652) = 0,29806736$

i_1, i_2, \dots, i_5 melakukan proses sama sesuai dengan persamaan diatas.

Tabel 11 Hasil proses lapisan gerbang input (i_t)

	i_1	i_2	i_3	i_4	i_5
i_t	0,29807	0,7031	0,7007	0,2999	0,7041

- $\check{C}_1 = \tanh(0,245) = 0,240050366$

$\check{C}_1, \check{C}_2, \dots, \check{C}_5$ melakukan proses sama sesuai dengan persamaan diatas.

Tabel 12 Hasil proses lapisan tanh candidate gate

	\check{C}_1	\check{C}_2	\check{C}_3	\check{C}_4	\check{C}_5
\check{C}_t	0,2401	-0,23666	-0,25633	0,269314	-0,2435

Gerbang pembuatan vektor baru = $i_t * \check{C}_t$

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

- $C_1 = (0,55207 * 0) + (0,29807 * 0,2401) = 0,071551$

C_1, C_2, \dots, C_5 melakukan proses sama sesuai dengan rumus diatas.

Tabel 13 Hasil proses Keadaan penghubung (state cell)

	C_1	C_2	C_3	C_4	C_5
C_t	0,0715	-0,1342	-0,239	-0,0536	-0,1954

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

$$o_1 = \sigma(-0,87038) = 0,295175002$$

o_1, o_2, \dots, o_5 melakukan proses sama sesuai dengan rumus diatas.

Tabel 14 Hasil proses Gerbang output (*output gate*)

	o_1	o_2	o_3	o_4	o_5
o_t	0,295	0,7003	0,7155	0,27708	0,701

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

C_t : didapat dari table 13

$$h_1 = \tanh(0,0715) = 0,0211$$

h_1, h_2, \dots, h_5 melakukan proses sama sesuai dengan rumus diatas.

Tabel 15 Hasil proses Block output (h_t)

	h_1	h_2	h_3	h_4	h_5
h_t	0,0211	-0,0934	-0,093	-0,1678	-0,015

Fase Propagasi maju (forward) hasil dari *Block output* (h_t) sama dengan hasil urutan berikutnya (prediksi) dari proses *Block output* (h_{t-1}). Jadi h_1, h_2, \dots, h_t sama dengan $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_t$.

x_t = data asli

\hat{y}_t = Prediksi dari data x_{t-1}

$\hat{y}_t = h_{t-1}$

Dari hasil *Block output* (h_t) 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. *Output block* terakhir (h_5) adalah output yang sesungguhnya atau hasil prediksi dari urutan - urutan sebelumnya.

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

Berdasarkan kesalahan ($y_t - \hat{y}_t$) dihitung *error output* (δh^t) 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 menerapkan data pada table 5 dan dari proses lanjutan dari fase forward :

Dari fase forward, didapat $y_t, W, h_{t-1}, I_t, z_t, f_t, (i_t, \check{C}_t), o_t, C_t, h_t$. Maka dalam fase ini menghitung *error* dari $\delta y_t, \delta W, \delta h_{t-1}, \delta I_t, \delta z_t, \delta f_t, (\delta i_t, \delta \check{C}_t), \delta o_t, \delta C_t, \delta h_t$ dan t dimulai dari angka 5 karena 5 proses urutan paling akhir.

1. *Error Output layer* (δy_t).

Dari proses forward didapat *block Output layer* atau output prediksi dimulai dari urutan paling terakhir yaitu (h_5) (h_5, \dots, h_2, h_1) dapat dilihat pada table 15 dan untuk membandingkan dengan data aktual dapat dilihat pada table 5. $E_0 = 0$

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

$$\begin{aligned} h_t &= \check{y}_t \\ y_t &= \text{data asli} \\ h_t &= \text{data prediksi dari data } y_{t+1} \\ E_t &= \frac{1}{2} \times \sum_{t=1}^n (y_t - \check{y}_t) \end{aligned} \quad (2.38)$$

Tabel 16 Hasil proses Menghitung Error dari data real dengan prediksi

	E_5	E_4	E_3	E_2	E_1
E_t	0,011462155	0,018878615	0,11186915	0,171721881	0,172162946

2. *Error block Output layer* (δh_t).

Langkah 2 Membandingkan antara y_{t+1} dan h_t jika $y_{t+1} > h_t = 0$ jika $y_{t+1} < h_t = 1$:

Tabel 17 Perbandingan y_{t+1} lebih besar dari h_t

	$y_{5+1} > h_5$	$y_{4+1} > h_4$	$y_{3+1} > h_3$	$y_{2+1} > h_2$	$y_{1+1} > h_1$
1/0	1	1	1	1	0

Tabel 18 Perbandingan y_{t+1} lebih besar dari h_t

	$y_{5+1} < h_5$	$y_{4+1} < h_4$	$y_{3+1} < h_3$	$y_{2+1} < h_2$	$y_{1+1} < h_1$
1/0	0	0	0	0	1

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

Error block Output layer (δh_{5+1}) awal sama dengan 0 karena proses dari *Error block Output layer* sebelumnya belum ada dan δC_{5+1} juga 0. δh_{t+1} (lama) sama dengan Proses *Error block Output layer* sebelumnya.

Tabel 19 Hasil proses block Output layer (δh_{t+1}) backward

	δh_{5+1}	δh_{4+1}	δh_{3+1}	δh_{2+1}	δh_{1+1}	δh_{0+1}
δh^{t+1}	0,000	-0,051	-0,016	-0,071	-0,086	-0,012

a. Menghitung *error block Output layer* dengan $y_{t+1} > h_t$

$$\delta \hat{h}_t(\text{baru}) = \delta h_{t+1} - 0,5 * (y_{t+1} > h_t) \dots\dots\dots(2.39)$$

$$\delta \hat{h}_5(\text{baru}) = \delta h_{5+1} - 0,5 * (y_{5+1} > h_5)$$

$$\delta \hat{h}_5(\text{baru}) = 0 - 0,5 * (1) = -0,5$$

Table 20 Proses *error block Output layer* $\delta \hat{h}_t$ dengan $y_{t+1} > h_t$

	δh_5	δh_4	δh_3	δh_2	δh_1
$\delta \hat{h}^t$	-0,5000	-0,5507	-0,5159	-0,5708	-0,0860

b. Menghitung *error block Output layer* dengan $y_{t+1} < h_t$

$$\delta h_t = \delta \hat{h}_t + 0,5 * (y_{t+1} < h_t) \dots\dots\dots(2.40)$$

$$\delta h_5 = \delta \hat{h}_5 + 0,5 * (y_{5+1} < h_5)$$

$$\delta h_5 = -0,5 + 0,5 * (0) = -0,5$$

Tabel 21 Proses *error block Output layer* δh_t dengan $y_{t+1} < h_t$

	δh_5	δh_4	δh_3	δh_2	δh_1
δh^t	-0,5000	-0,5507	-0,5159	-0,5708	0,4140

$\delta h_5, \delta h_4, \dots, \delta h_1$ melakukan proses sama sesuai dengan rumus diatas.

3. Pada proses mencari (h_t) pada forward ditemukan persamaan (2,12) maka langkah selanjutnya menghitung δo_t dan δC_t

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

t menunjukkan pada urutan dimulai dari 5

$$\delta o_t = \delta h_t \odot \tanh(C_t) \dots\dots\dots(2.18)$$

C_t = dapat dilihat pada *Table 13*,

$$\delta o_5 = -0,5 \odot \tanh(-0,1954) = 0,09647763$$

Tabel 22 Proses *error Error Output gate* (δo_t)

	δo_5	δo_4	δo_3	δo_2	δo_1
δo^t	0,096	0,029	0,121	0,076	0,030

$\delta o_5, \delta o_4, \dots, \delta o_1$ melakukan proses sama sesuai dengan rumus diatas.

Langkah 4 dan langkah 5 adalah proses *Output gate*

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

Tabel 23 Proses *Error Output gate* sebelumnya (δC_{t+1})

	δC_{5+1}	δC_{4+1}	δC_{3+1}	δC_{2+1}	δC_{1+1}
δC^{t+1}	0	-0,337	-0,303	-0,519	-0,622

$$\delta C_t = \left(\delta h_t \odot \delta o_t \odot \left(1 - \tanh^2(C_t) \right) \right) + \delta C_{t+1} \dots\dots\dots(2.19)$$

$$\delta C_5 = \left(\delta h_5 \odot \delta o_5 \odot \left(1 - \tanh^2(C_5) \right) \right) + \delta C_{5+1}$$

$$\delta C_5 = \left(-0,1352 \odot 0,70092 \odot \left(1 - \tanh^2(-0,1954) \right) \right) + 0$$

$$\delta C_5 = -0,337411626$$

Tabel 24 Proses Menghitung *Error cell state* (δC_t)

	δC_5	δC_4	δC_3	δC_2	δC_1
δC^t	-0,337	-0,303	-0,519	-0,622	-0,158

$\delta C_5, \delta C_4, \dots, \delta C_1$ melakukan proses sama sesuai dengan rumus diatas.

4. Pada proses forward diberikan persamaan (2.9) dan ditemukan

$$\delta i_t, \delta a_t, \delta f_t, \delta C_{t-1}$$

a^t : *candidate gate* (\check{C}_t)

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

$$\delta i_t = \delta C_t \odot a_t \dots \dots \dots (2.20)$$

$$\delta i_5 = \delta C_5 \odot a_5$$

$$\delta i_5 = -0,337 \odot -0,1954 = 0,08214642$$

Langkah 7 Hitung informasi *error* gerbang *candidate connecting state* a^t

$$\delta a_t = \delta C_t \odot i_t \dots \dots \dots (2.21)$$

$$\delta a_5 = \delta C_5 \odot i_5$$

$$\delta a_5 = -0,337 \odot 0,7041 = -0,237567466$$

Langkah 6 dan langkah 7 adalah proses *write gate*

Langkah 8 Hitung informasi *error* gerbang *forget gate* δf^t

$$\delta f_t = \delta C_t \odot C_{t-1} \dots \dots \dots (2.22)$$

$$\delta f_5 = \delta C_5 \odot C_{5-1}$$

$$\delta f_5 = -0,337 \odot -0,0536 = 0,018080133$$

Langkah 8 adalah proses *Keep gate*

Langkah 9 Hitung informasi *error* Keadaan penghubung sebelumnya δc^{t-1}

$$\delta C_{t-1} = \delta C_t \odot f_t \dots \dots \dots (2.23)$$

$$\delta C_{t-1} = \delta C_5 \odot f_5$$

$$\delta C_{t-1} = -0,337 \odot 0,44765 = -0,151042409$$

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

Table 25 proses Menghitung $\delta i_t, \delta a_t, \delta f_t, \delta C_{t-1}$

t	5	4	3	2	1
δi_t	0,082	-0,082	0,133	0,147	-0,038
δa_t	-0,238	-0,091	-0,364	-0,437	-0,047
δf_t	0,018	0,072	0,070	-0,045	0,000
δC_{t-1}	-0,151	-0,170	-0,230	-0,280	-0,087

Untuk t_5, \dots, t_2, t_1 dalam mencari $\delta i_t, \delta a_t, \delta f_t, \delta C_{t-1}$ dilakukan proses sama sesuai dengan persamaan diatas. δC_{t-1} nantinya akan dibawa ke *hidden layer* sebelumnya (δC_{t+1}) dalam *table 23*.

5. Pada proses forward diberikan persamaan (2.24) dan ditemukan δz^t maka menghitung $\delta z a^t, \delta z i^t, \delta z f^t, \delta z o^t$ dan menghasilkan δz^t :

Langkah 10 Hitung *error block input candidate gate* ($\delta z a^t$)

$$\hat{a}^t = z_t^{\check{c}_1} \text{ (Terdapat pada table 9)}$$

$$\delta z \hat{a}_t = \delta a_t \odot (1 - \tanh^2(\hat{a}_t)) \dots\dots\dots(2.24)$$

$$\delta z \hat{a}_5 = \delta a_5 \odot (1 - \tanh^2(\hat{a}_5))$$

$$\delta z \hat{a}_5 = -0,2376 \odot (1 - \tanh^2(-0,2418)) = -0,223486118$$

Langkah 11 Hitung *error block input input gate* ($\delta z i^t$)

$$\delta z \hat{i}_t = \delta i_t \odot i_t \odot (1 - i_t) \dots\dots\dots(2.25)$$

$$\delta z \hat{i}_5 = \delta i_5 \odot i_5 \odot (1 - i_5)$$

$$\delta z \hat{i}_5 = 0,082 \odot 0,7041 \odot (1 - 0,7041) = 0,017115052$$

Langkah 12 Hitung *error block input forget gate* ($\delta z f^t$)

$$\delta z \hat{f}_t = \delta f_t \odot f_t \odot (1 - f_t) \dots\dots\dots(2.26)$$

$$\delta z \hat{f}_5 = \delta f_5 \odot f_5 \odot (1 - f_5)$$

$$\delta z \hat{f}_t = 0,0181 \odot 0,44765 \odot (1 - 0,44765) = 0,004470485$$

Langkah 13 Hitung *error block input output gate* ($\delta z o^t$)

$$\delta z \hat{o}_t = \delta o_t \odot o_t \odot (1 - o_t) \dots\dots\dots(2.27)$$

$$\delta z \hat{o}_5 = \delta o_5 \odot o_5 \odot (1 - o_5)$$

$$\delta z \hat{o}_5 = 0,0965 \odot 0,701 \odot (1 - 0,701) = 0,020224728$$

Langkah 14 Menggabungkan langkah 10, 11, 12,13 menjadi (δz^t)

Tabel 26 Hasil proses $\delta z a^t, \delta z i^t, \delta z f^t, \delta z o^t$ untuk block input (δz^t)

T	δz^5	δz^4	δz^3	δz^2	δz^1
$\delta z \hat{a}_t$	-0,223	-0,223	-0,223	-0,223	-0,223
$\delta z \hat{i}_t$	0,017	0,017	0,017	0,017	0,017
$\delta z \hat{f}_t$	0,004	0,004	0,004	0,004	0,004
$\delta z \hat{o}_t$	0,020	0,020	0,020	0,020	0,020

Dari langkah 14 didapat *block input* (δz^t).

Untuk t_5, \dots, t_2, t_1 dalam mencari δz_t dilakukan proses sama sesuai dengan rumus diatas sampai t_1 .

6. Dari tahap *forward pass* diberikan persamaan (2.17) dan ditemukan δz^t dari langkah sebelumnya untuk dicari δW^t dan δh^{t-1} .

Langkah 15

Tahap $\delta z^{t^T} = [\delta z \hat{a}^t, \delta z \hat{i}^t, \delta z \hat{f}^t, \delta z \hat{o}^t]^T$ untuk menghitung δI^t

$$\delta z^{t^T} = \begin{bmatrix} \delta z \hat{a}^t \\ \delta z \hat{l}^t \\ \delta z \hat{f}^t \\ \delta z \hat{o}^t \end{bmatrix} \dots\dots\dots(2.29)$$

$$\delta I^t = \delta z^t \cdot W^T \dots\dots\dots(2.31)$$

$$\delta I^5 = \begin{bmatrix} -0,223 \\ 0,017 \\ 0,0045 \\ 0,020 \end{bmatrix} \cdot \begin{bmatrix} -0,2457 & 0,868 & -0,2064 & 0,843 \\ 0,029 & -0,3795 & -0,085 & 0,9077 \\ 0,184 & 0,0795 & 0,251 & -0,594 \end{bmatrix}$$

$$\delta I^5 = [0,085893946 \quad 0,004941116 \quad 0,050735197]$$

Tabel 27 Hasil proses δI^t untuk menghasilkan δh^{t-1}

t	δI^5	δI^4	δI^3	δI^2	δI^1
δb	0,0859	0,0071	0,1250	0,1439	0,0092
$\delta x^{t=1}$	0,0049	0,0079	0,0003	-0,0083	0,0073
δh^{t-1}	-0,0507	-0,0159	-0,0708	-0,0860	-0,0125

Dari proses ini didapat δh_{t-1} akan menghubungkan ke layer sebelumnya (δh_{t+1}) dan tambahkan ke tabel 19 pada kolom δh_{t+1} , proses ini mengulangi terus sampai $t = 1$ awal

7. Menghitung Derivatif Bobot δW^t

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

$$\delta W^t = \delta z^t \cdot I^{t^T} \dots\dots\dots(2.33)$$

$$\delta W^t = \begin{bmatrix} W^{\tilde{c}} \cdot \delta z \hat{a}^t \\ W^i \cdot \delta z \hat{l}^t \\ W^f \cdot \delta z \hat{f}^t \\ W^o \cdot \delta z \hat{o}^t \end{bmatrix} \cdot [W^{\tilde{c}} \cdot \delta z \hat{a}^t, W^i \cdot \delta z \hat{l}^t, W^f \cdot \delta z \hat{f}^t, W^o \cdot \delta z \hat{o}^t] \dots\dots(2.34)$$

$$\delta W^5 = \delta z^5 \cdot I^{5^T}$$

$$\delta W^5 = \begin{bmatrix} -0,223 \\ 0,017 \\ 0,0045 \\ 0,020 \end{bmatrix} \cdot [0,0859 \quad 0,0049 \quad -0,0507]$$

$$\delta W^5 = \begin{bmatrix} -0,223 & -0,004 & 0,03 \\ 0,017 & 0,0003 & -0,002 \\ 0,0045 & 0,000007 & -0,0006 \\ 0,0202 & 0,0003 & -0,0027 \end{bmatrix}$$

Tabel 28 Hasil Derivatif Bobot δW^t

t	δW^t		
δW^5	-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
δW^4	-0,08433	0	0,001251
	-0,01714	0	0,000254
	0,017834	0	-0,00026
	0,005906	0	-8,8E-05
δW^3	-0,33997	-0,006181	0,057047
	0,027917	0,000507579	-0,00468
	0,017195	0,000312629	-0,00289
	0,024631	0,000447845	-0,00413
δW^2	-0,41299	-0,008343	-0,00871
	0,030742	0,000621048	0,000648
	-0,01102	-0,000222	-0,00023
	0,015985	0,000322927	0,000337
δW^1	-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 :

$$\delta W^t(\text{baru}) = \delta W^t(\text{lama}) + \delta W^{t+1} \dots \dots \dots (2.35)$$

Tabel 29 Modifikasi bobot untuk proses optimasi

t	δW^t (lama)			δW^{t+1}			δW^t (baru)		
δW^5	-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
δW^4	-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
δW^3	-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
δW^2	-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
δW^1	-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

C. Fase ketiga : Perubahan bobot optimasi bobot (*final bobot*)

Setelah semua *error output* (δh^t) 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.

$W_t(\text{Lama}) = \text{terdapat pada table 7}$

$\text{learning_rate } (a) = 0,2$

$$W_t(\text{Baru}) = W_t(\text{Lama}) - \text{learning_rate} * \delta W^1 \dots \dots \dots (2.37)$$

Tabel 30 Bobot yang sudah di update ($W_t(\text{Baru})$)

t	δW^t			$a * \delta W^1$			$W_t(\text{Baru})$		
W_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 ditunjukkan di *table 3* menghasilkan *update* bobot baru $W_t(\text{Baru})$ 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*.

3.7.2 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 menerapkan data pada table 4 (data *testing*) :

1. Input Layer
2. Inisialisasi bobot $W_t(\text{Baru})$ yang didapat dari proses training sebelumnya.

Tabel 31 Data sample testing

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

- Didapat bobot $W_t(\text{Baru})$ dari proses sebelumnya, yaitu proses *training* :

Tabel 32 Bobot baru $W_t(\text{baru})$

W^f	-0,025	0,0133	0,1844
W^i	0,85789	-0,37837	0,0795
$W^{\check{c}}$	-0,212	-0,084	0,251
W^o	0,828	0,9090	-0,5937

5. Persiapan block input (z_t)
 - Penggabungan data sample tranpose dengan *block output* (h^{t-1}) dengan menggunakan persamaan (2.13)

Tabel 33 Penggabungan data sample tranpose dengan *block output* (h^{t-1})

t	I^1	I^2	I^3	I^4
b	1	1	1	1
$x^{t=1}$	0,0161	0,0262	0,022	0,0242
h^{t-1}	0,0000	-0,012	-0,019	-0,022

h^{t-1} sama dengan 0 dikarenakan hasil *block output* dari proses hidden layer sebelumnya (h^{t-1}) belum ada, dan untuk (h^{2-1}) adalah hasil block output dari proses hidden layer urutan sebelumnya dan seterusnya.

- Perkalian antara I^t dengan bobot W^f , W^i , $W^{\check{c}}$, W^o dengan menggunakan persamaan (2.15)

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

z_t	z_1	z_2	z_2	z_3	z_4
z_t^f	-0,0245	-0,0265	-0,027	-0,0284	-0,0245
z_t^i	0,8518	0,847	0,85	0,847	0,852
$z_t^{\check{c}1}$	-0,2135	-0,217	-0,219	-0,22	-0,2135
z_t^o	0,843	0,86	0,859	0,8633	0,843

Setelah didapat block input (z_1) 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

6. Hidden layer

Langkah 1 Gerbang lupa (*forget gate*):

$$f_t = \sigma(z_t^f) \dots\dots\dots(2.3)$$

$$f_1 = \sigma(-0,0245) = 0,446825469$$

f_1, f_2, \dots, f_4 melakukan proses sama sesuai dengan rumus diatas.

Tabel 35 Hasil proses Gerbang lupa (f_t)

	f_1	f_2	f_3	f_4
f_t	0,4468	0,446	0,445	0,4453

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

- $i_t = \sigma(z_t^i) \dots\dots\dots(2.6)$

$$i_1 = \sigma(0,8518) = 0,700937789$$

i_1, i_2, \dots, i_4 melakukan proses sama sesuai dengan persamaan diatas.

Tabel 36 Hasil proses lapisan gerbang input (i_t)

	i_1	i_2	i_3	i_4
i_t	0,7009	0,69	0,7	0,69

$$4 \quad \check{c}_t = \tanh(z_t^{\check{c}}) \dots\dots\dots(2.8)$$

$$\check{c}_1 = \tanh(-0,2135) = -0,024458074$$

$\check{c}_1, \check{c}_2, \dots, \check{c}_4$ melakukan proses sama sesuai dengan persamaan diatas.

Tabel 37 Hasil proses lapisan *tanh candidate gate*

	\check{c}_1	\check{c}_2	\check{c}_3	\check{c}_4	\check{c}_5
\check{c}_t	-0,02445	-0,23666	-0,25633	0,269314	-0,2435

Gerbang pembuatan vektor baru = $i_t * \check{c}_t$

Langkah 3 Keadaan penghubung (*state cell*) :

$$C_t = f_t * c_{t-1} + i_t * \check{c}_t \dots\dots\dots(2.9)$$

$$C_1 = (0,4468 * 0) + (0,701 * -0,2401) = -0,017143588$$

C_1, C_2, \dots, C_4 melakukan proses sama sesuai dengan persamaan diatas.

Tabel 38 Hasil proses Keadaan penghubung (*state cell*)

	C_1	C_2	C_3	C_4
C_t	-0,0171	-0,0265	-0,278	-0,028

Langkah 4 Gerbang output (*output gate*) :

$$o_t = \sigma(z_t^o) \dots\dots\dots(2.11)$$

$$o_1 = \sigma(0,843) = 0,699093763$$

o_1, o_2, \dots, o_4 melakukan proses sama sesuai dengan persamaan diatas.

Tabel 39 Hasil proses Gerbang output (*output gate*)

	o_1	o_2	o_3	o_4
o_t	0,699	0,7025	0,7025	0,70335

Langkah 5 Block output

$$h_t = o_t \cdot \tanh(c_t) \dots\dots\dots(2.12)$$

C_t : didapat dari table 13

$$h_1 = 0,699 * \tanh(-0,0171) = -0,011983802$$

h_1, h_2, \dots, h_4 melakukan proses sama sesuai dengan persamaan diatas.

Tabel 40 hasil proses *Block output* (h_t)

	h_1	h_2	h_3	h_4
h_t	-0,0198	-0,0184	-0,022	-0,0237

Fase Propagasi maju (forward) hasil dari *Block output* (h_t) sama dengan hasil urutan berikutnya (prediksi) dari proses *Block output* (h_{t-1}). Jadi h_1, h_2, \dots, h_t sama dengan $\hat{y}_{1+1}, \hat{y}_{2+1}, \dots, \hat{y}_{t+1}$.

7. Denormalisasi (menjadikan hasil prediksi)

$$x_i = f(x) * x_{max} + x_{min} \dots\dots\dots(2.41)$$

$$h_4 = -0,0237$$

$$h_4 = f(x)$$

$$x_i = -0,0237 * 12375 + 12000$$

$$x_i = 11706,59$$

Jadi hasil prediksi hari ke lima dari proses prediksi (\hat{y}_5) diatas adalah 11706,59 .

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

y_i dapat dilihat di table 5

$$E = y_5 - \hat{y}_5$$

$$E = 12000 - 11706,59$$

$$E = 293,41$$

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

3.8 Skenario Uji Coba dan Evaluasi

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

BAB IV IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan langkah–langkah pengerjaan sistem prediksi harga saham menggunakan metode *LSTM*, yang di implementasikan ke dalam bahasa pemrograman python, sebelum sistem dilepas ke user terlebih dahulu harus dipastikan apakah sitem telah sesuai dengan tujuan perancangan sistem, oleh karena itu, pada bab ini bertujuan untuk memastikan hasil rancangan sistem yang telah dibentuk sesuai. Langkah ini bertujuan untuk mengetahui apakah sistem telah memenuhi kebutuhan. Kemudian dilakukan analisis untuk menentukan keakuratan sistem dalam mengolah data.

4.1 Lingkungan Uji Coba

Pada bagian ini diuraikan tentang lingkungan uji coba, yaitu meliputi perangkat keras dan perangkat lunak untuk menjalankan program yang sudah dibuat. Lingkungan uji coba ini dilakukan dengan bantuan *VisualStudio Code* . Lingkungan Uji Coba dapat dilihat pada Tabel 4.1.

Tabel 42 Lingkungan Uji Coba

No	Kebutuhan	Jenis
1	Operating sistem	Windows 10 Pro
2	<i>Program Tool</i>	IDLE Python
3	<i>Framework Python</i>	Dash, Pandas, Yfinance, Numpy.
4	<i>Text Editor</i>	Visual studio code

4.2 Source Code Objek LSTM

Pada tahap ini akan diimplementasikan *Source code* dari proses forward, backward, training, dan testing.

```
class LSTMCell:

    def __init__(self, inputSize, numCells, bobot):
        self.inputSize = inputSize
        self.numCells = numCells
        self.W = bobot # Matrix bobot

        # tampungan inisial
        self.h = []
        self.C = []
        self.C_bar = []
        self.i = []
        self.f = []
        self.o = []

        self.I = []
        self.z = []

    def forwardStep(self, x, jenis):
        if jenis == "prediksi":
            models = pd.read_csv("model.csv")
            model = models.values
        else:
            model = self.W
        I = np.concatenate((x, self.h[-1]))
        self.I.append(I)
        z = np.dot(model, I)
        self.z.append(z)

        C_bar = np.tanh(z[0:self.numCells])
        self.C_bar.append(C_bar)

        i = sigmoid(z[self.numCells:self.numCells * 2])
        self.i.append(i)

        f = sigmoid(z[self.numCells * 2:self.numCells * 3])
        self.f.append(f)

        o = sigmoid(z[self.numCells * 3:])
        self.o.append(o)

        C = np.multiply(f, self.C[-1]) + np.multiply(i, C_bar)
        self.C.append(C)
```

```

        h = np.multiply(o, np.tanh(C))
        self.h.append(h)
        return (h,C,o,f,i,C_bar,z,I,model)

# Forward antar block
def forwardPass(self, x,jenis):
    self.h = []
    self.C = []
    self.C_bar = []
    self.i = []
    self.f = []
    self.o = []
    self.I = []
    self.z = []

    numCells = self.numCells
    self.h.append(np.zeros(numCells))
    self.C.append(np.zeros(numCells))
    self.C_bar.append(np.zeros(numCells))
    self.i.append(np.zeros(numCells))
    self.f.append(np.zeros(numCells))
    self.o.append(np.zeros(numCells))
    self.I.append(np.zeros(numCells))
    self.z.append(np.zeros(numCells))

    O_h = [self.forwardStep(x_t,jenis)[0] for x_t in x]

    return (O_h)

def backwardStep(self, t, dE_dh_t, dE_dc_tplus1):

    dE_do_t = np.multiply(dE_dh_t, np.tanh(self.C[t]))

    dE_dc_t_1= (np.ones(self.numCells)-np.square(np.tanh(self.C[t])))
    dE_dc_t_2= np.multiply(np.multiply(dE_dh_t,self.o[t]),dE_dc_t_1)
    dE_dc_t = dE_dc_tplus1 + dE_dc_t_2

    dE_di_t = np.multiply(dE_dc_t, self.C_bar[t])
    dE_dcbar_t = np.multiply(dE_dc_t, self.i[t])
    dE_df_t = np.multiply(dE_dc_t, self.C[t - 1])
    dE_dc_tminus1 = np.multiply(dE_dc_t, self.f[t])

    dE_dzcbar_t_1 = np.square(np.tanh(self.z[t][0:self.numCells]))
    dE_dzcbar_t_2 = (np.ones(self.numCells) - dE_dzcbar_t_1)
    dE_dzcbar_t = np.multiply(dE_dcbar_t, dE_dzcbar_t_2)

    dE_dzi_t_1 = (np.ones(self.numCells) - self.i[t])
    dE_dzi_t = np.multiply(np.multiply(dE_di_t, self.i[t]),dE_dzi_t_1)

```

```

dE_dzf_t_1 = (np.ones(self.numCells) - self.f[t])
dE_dzf_t = np.multiply(np.multiply(dE_df_t, self.f[t]), dE_dzf_t_1)

dE_dzo_t_1 = (np.ones(self.numCells) - self.o[t])
dE_dzo_t = np.multiply(np.multiply(dE_do_t, self.o[t]), dE_dzo_t_1)

dE_dz_t = np.concatenate((dE_dzcbar_t, dE_dzi_t, dE_dzf_t, dE_dzo_t))

dE_dI_t = np.dot(np.transpose(self.W), dE_dz_t)

dE_dh_tminus1 = dE_dI_t[self.inputSize:]

dE_dz_t.shape = (len(dE_dz_t), 1)
self.I[t].shape = (len(self.I[t]), 1)
dE_dW_t = np.dot(dE_dz_t, np.transpose(self.I[t]))

return (dE_dW_t, dE_dh_tminus1, dE_dc_tminus1)

# Back propagation through time
def BPTT(self, y):
    numTimePeriods = len(y)
    dE_dW = 0
    dE_dh_t = 0
    dE_dc_t = 0
    E = 0.0
    discount = 1.0

    for i in range(numTimePeriods):
        index = numTimePeriods - i
        E = E + 0.5 * np.sum(np.absolute(self.h[index] - y[index - 1]))

    lessThan = np.less(self.h[index], y[index - 1])
    greaterThan = np.greater(self.h[index], y[index - 1])
    dE_dh_t -= 0.5 * lessThan
    dE_dh_t += 0.5 * greaterThan

    result = self.backwardStep(index, dE_dh_t, dE_dc_t)
    dE_dW = dE_dW + result[0]

    dE_dh_t = result[1]
    dE_dc_t = result[2]

    discount *= 0.99

    return (E / (numTimePeriods), dE_dW)
)

```

```

def train(self,trainingData,numEpochs,learningRate
        ,sequenceLength,max_ex,min_ex):

    adaptiveLearningRate = learningRate
    error_t=[]

    for epoch in range(numEpochs):
        trainingSequences=sequenceProducer(trainingData,sequenceLength
)

        epochError = 0.0
        counter = 0

        for sequence in trainingSequences:
            counter += 1

            self.forwardPass(sequence[:],"no_prediksi")

            result = self.BPTT(sequence[:,2:])

            E = result[0]
            dE_dW = result[1]
            w = dE_dW.shape

            adaptiveLearningRate = learningRate / (1 + (epoch/10))
            self.W = self.W - adaptiveLearningRate * dE_dW
            optimasi = [[self.W]]

            epochError += E
            error_t.append([epoch,epochError / counter])
        tbl_error = pd.DataFrame(data = error_t,columns=["urutan","error"]
)

        tbl_error.to_csv("tbl_error.csv")

        model = pd.DataFrame(self.W)
        model.to_csv("model.csv",index=False)
        print ("tbl_error ",tbl_error)

        return (tbl_error)

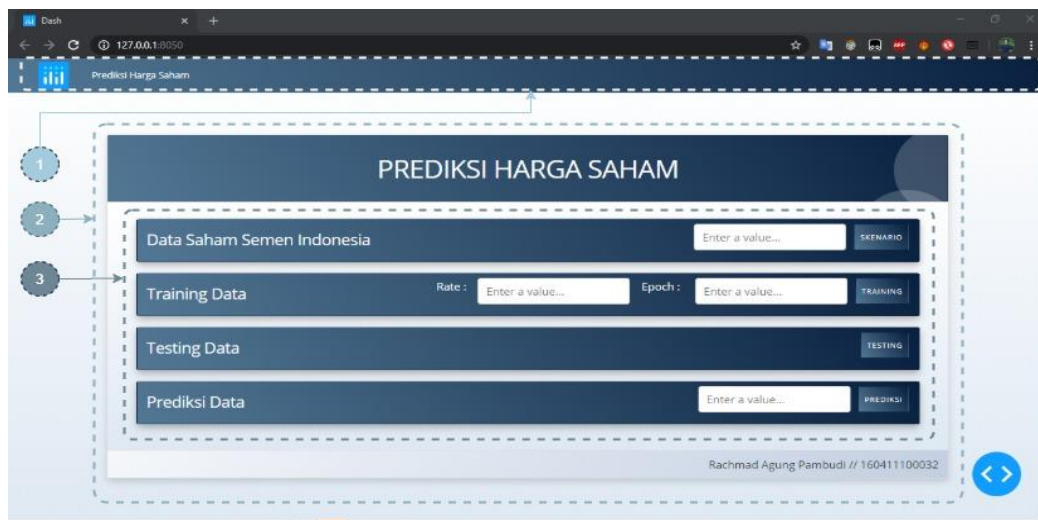
def forecast(self, forecastingData):
    self.forwardPass(forecastingData,"prediksi")
    f_h = np.transpose(np.transpose(self.h[-1]))
    return (f_h)

```

4.3 Graphical User Interface (GUI)

Pada penelitian ini dibuatkan suatu graphical user interface (GUI) yang berisi komponen – komponen sederhana yang akan mempermudah dalam mengetahui prediksi harga saham, pada pembuatan GUI pada sistem ini, data training akan di uji dengan 1 data testing, jadi untuk tampilan pada sistem ini hanya bisa melihat dan mengklasifikasikan 1 prediksi harga saham.

4.3.1 Tampilan Aplikasi



Gambar 4.21 Tampilan Awal Aplikasi

Dari *Gambar 4.1* tampilan awal aplikasi ini diberikan 3 bagian yaitu:

1. Navbar atau judul dari aplikasi.
2. Pengelompokan fitur-fitur yang diberikan oleh aplikasi.
3. *Field* dan tombol untuk menjalankan fitur – fitur didalam aplikasi.

4.3.2 Tampilan Menu



Gambar 4.22 Tampilan Menu fitur – fitur

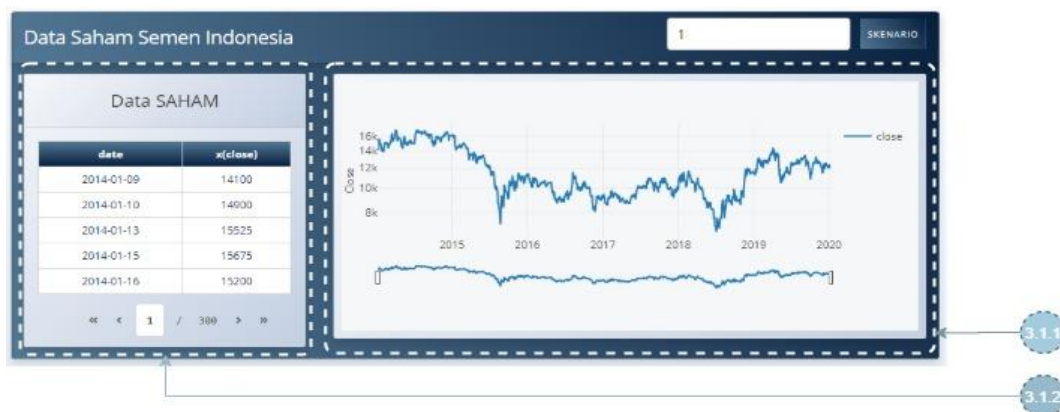
Dari *Gambar 4.2*. Tampilan Menu fitur – fitur ini dibagi menjadi 4 sesuai dengan fungsinya yaitu:

1. *Poin 3.1* diberikan 1 *Field* dan 1 tombol yang berfungsi untuk menyimpan variabel skenario berapa yang akan dieksekusi, dan akan menampilkan data tabel dan grafik harga saham.
2. *Poin 3.2* diberikan 2 *Field* dan 1 tombol yang berfungsi untuk menyimpan variabel learning rate dan epoch berapa yang akan digunakan, dan akan menampilkan data tabel dan grafik hasil training.
3. *Poin 3.3* diberikan 1 tombol yang berfungsi untuk mengeksekusi data testing dan akan menampilkan data testing tabel dan grafik hasil testing.
4. *Poin 3.1* diberikan 1 *Field* dan 1 tombol yang berfungsi untuk menyimpan variabel berapa hari yang akan diprediksi, dan akan menampilkan data tabel dan grafik hasil prediksi.

4.3.3 Tampilan fitur

4.3.3.1 Tampilan tabel dan grafik *full data close* saham

Dalam tampilan ini terdapat juga 1 *Field* dan 1 tombol yang berfungsi untuk menyimpan variabel skenario, ada 3 skenario yang dapat dijalankan yaitu skenario pertama 250 data training dan 250 data testing, skenario kedua 500 data training dan 500 data testing, dan skenario ketiga 1000 data training dan 500 data testing.



Gambar 4.23 Tampilan tabel dan grafik full data close saham

Dari *Gambar 4.3*. Tampilan tabel dan grafik full data close saham dibagi menjadi 2 yaitu:

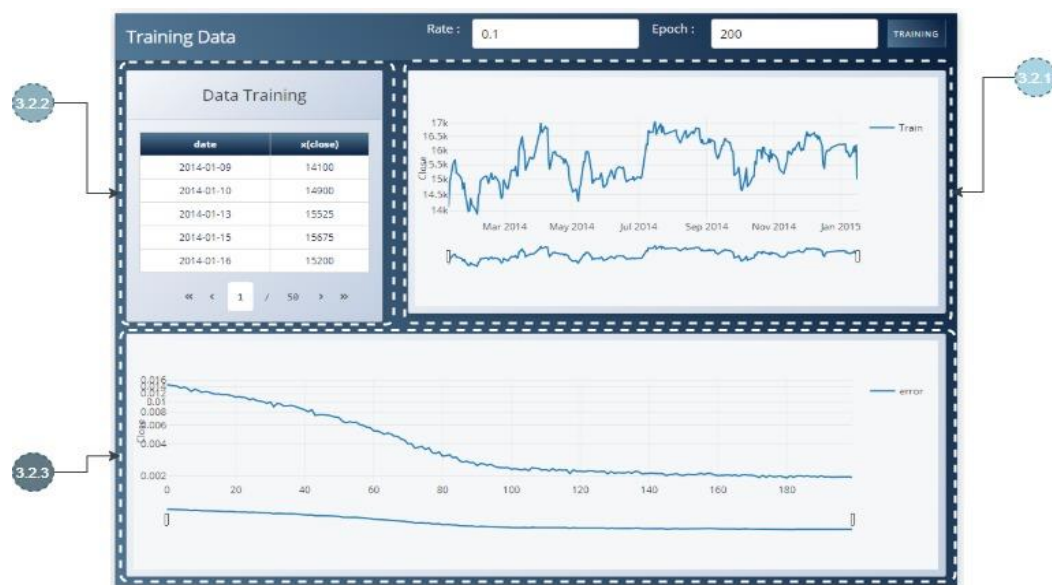
1. *Poin 3.1.1* diberikan tampilan tabel data yang berfungsi untuk memperlihatkan data saham seluruhnya.

2. Poin 3.1.2 diberikan tampilan grafik data yang berfungsi untuk memperlihatkan grafik data saham seluruhnya.

4.3.3.2 Tampilan Training data

Dalam tampilan ini juga terdapat 2 *Field* dan 1 tombol yang berfungsi untuk menyimpan variabel epoch dan learning rate, dan jika ditekan tombonya akan mengarah ke proses fungsi training.

Pada tampilan ini menjalankan fungsi *training* dalam proses *backend* nya berjalan dari proses *forward*, *backward*, *update* bobot, dan sampai optimasi. proses training ini akan menghasilkan bobot atau model baru yang sudah di *training* dan disimpan dalam file berformat csv, yang nantinya akan digunakan untuk proses testing maupun proses prediksi.

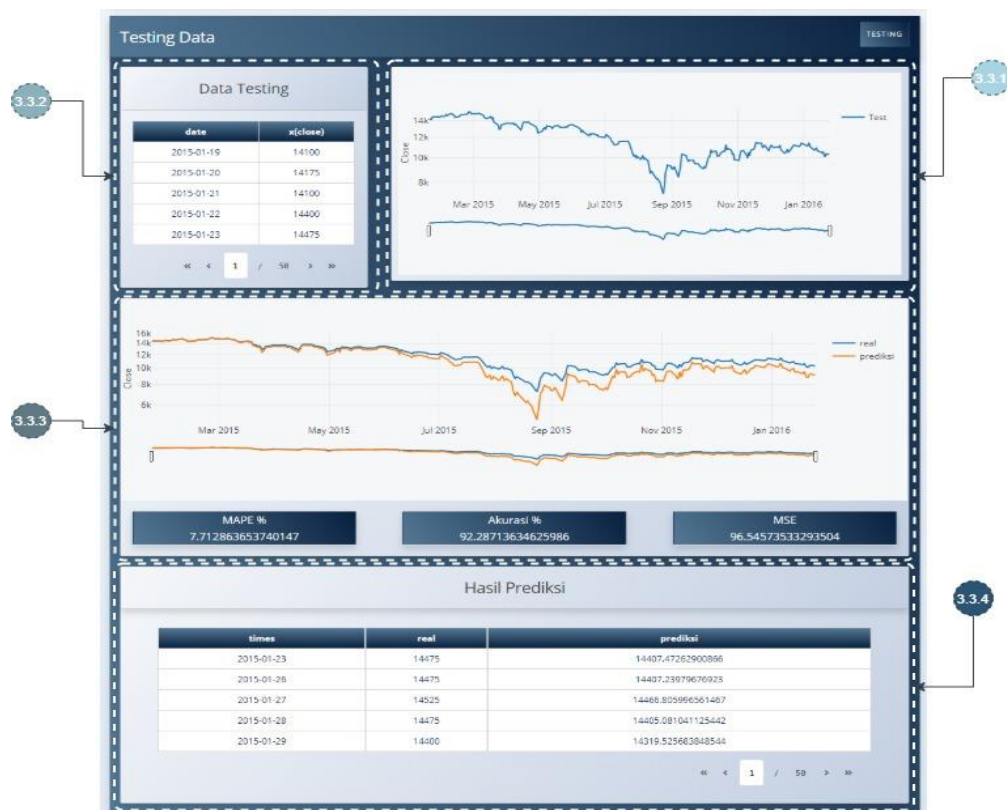


Gambar 4.24 Tampilan Training data

Dari Gambar 4.4 Tampilan Training data dibagi menjadi 3 konten sesuai dengan fungsinya yaitu:

1. Poin 3.2.1 diberikan tampilan tabel data yang berfungsi untuk memperlihatkan data training saja.
2. Poin 3.2.2 diberikan tampilan grafik data yang berfungsi untuk memperlihatkan grafik data training.
3. Poin 3.2.3 diberikan tampilan grafik hasil proses training yang menunjukkan tingkat errornya.

4.3.3.3 Tampilan Testing data



Gambar 4.25 Tampilan Testing Data

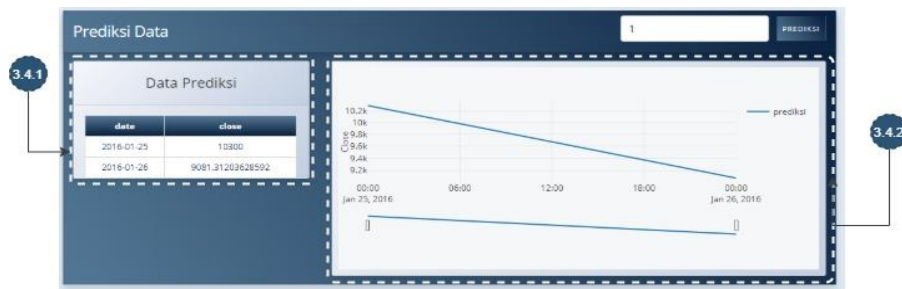
Dari Gambar 4.5. Tampilan Testing data dibagi menjadi 4 konten sesuai dengan fungsinya yaitu:

1. *Poin 3.3.1* diberikan tampilan tabel data yang berfungsi untuk memperlihatkan data *testing* saja.
2. *Poin 3.3.2* diberikan tampilan grafik data yang berfungsi untuk memperlihatkan grafik data *testing*.
3. *Poin 3.3.3* diberikan tampilan grafik hasil proses testing yang menunjukkan perbandingan hasil prediksi dengan nilai aktualnya, dan diperlihatkan juga hasil dari MAPE, Akurasi, MSE.
4. *Poin 3.3.4* diberikan tampilan tabel data yang berfungsi untuk memperlihatkan data hasil *testing*.

Pada saat tombol ditekan akan menunjukkan tampilan seperti gambar 5 yang akan menjalankan fungsi *testing* dalam proses *backend* nya memproses *forward* saja. proses testing ini akan menghitung juga hasil akurasi, MAPE, MSE, yang didapat dan tabel hasil prediksi dan nilai aktualnya.

4.3.3.4 Tampilan Prediksi data

Pada saat tombol ditekan akan menunjukkan tampilan seperti gambar 5 yang akan menjalankan fungsi prediksi dalam proses *backend* nya memproses *forward* saja.



Gambar 4.26. Tampilan tabel dan grafik prediksi data close saham

Dari *Gambar 4.6* Tampilan tabel dan grafik full data close saham dibagi menjadi 2 yaitu:

1. *Poin 3.4.1* diberikan tampilan tabel data yang berfungsi untuk memperlihatkan data prediksi.
2. *Poin 3.4.2* diberikan tampilan grafik data yang berfungsi untuk memperlihatkan grafik prediksi naik atau turun harganya.

4.3.4 Tampilan Full



Gambar 4.27. Tampilan full Aplikasi

4.4 Uji Coba Sistem

Uji coba sistem meliputi uji lags signifikan, uji scenario dan uji parameter dari setiap model data saham Semen Indonesia.

4.4.1 Uji Coba Skenario

Pada bagian skenario uji coba ini dijelaskan mengenai skenario ujicoba yang telah dilakukan. Uji coba dilakukan untuk membagi data. Dalam penelitian ini dilakukan 3 skenario percobaan. Tiap skenario akan dihitung akurasi MSE dan MAPE. Pada tiap skenario dibedakan berdasarkan susunan data training dan data testing.

Table 42 Skenario Uji Coba

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

Dari semua percobaan 3 skenario akan dihitung nilai MSE dan MAPE. Berikut akan adalah penjelasan dari masing-masing skenario.

Skenario 1

Table 43 Data training dan data testing skenario 1

Ujicoba	Trainig	Testing
1	250 DATA	250 DATA

Penjelasan dari tabel 4.2 adalah total data yang dipakai adalah 500 data saham, dari data tersebut dibagi lagi menjadi 250 data training dan 250 data testing. Dilakukan proses LSTM dengan melalui tahap sehingga memperoleh bobot baru Berikut hasil uji coba dari skenario 1.

Table 44 Perubahan Uji Coba dengan perubahan nilai epoch pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
5	0.2	1475.95	33.32	66.67
10	0.2	1307.54	31.35	68.65
25	0.2	978.08	27.28	72.72
50	0.2	411.52	17.89	82.11
100	0.2	5.51	1.85	98.15

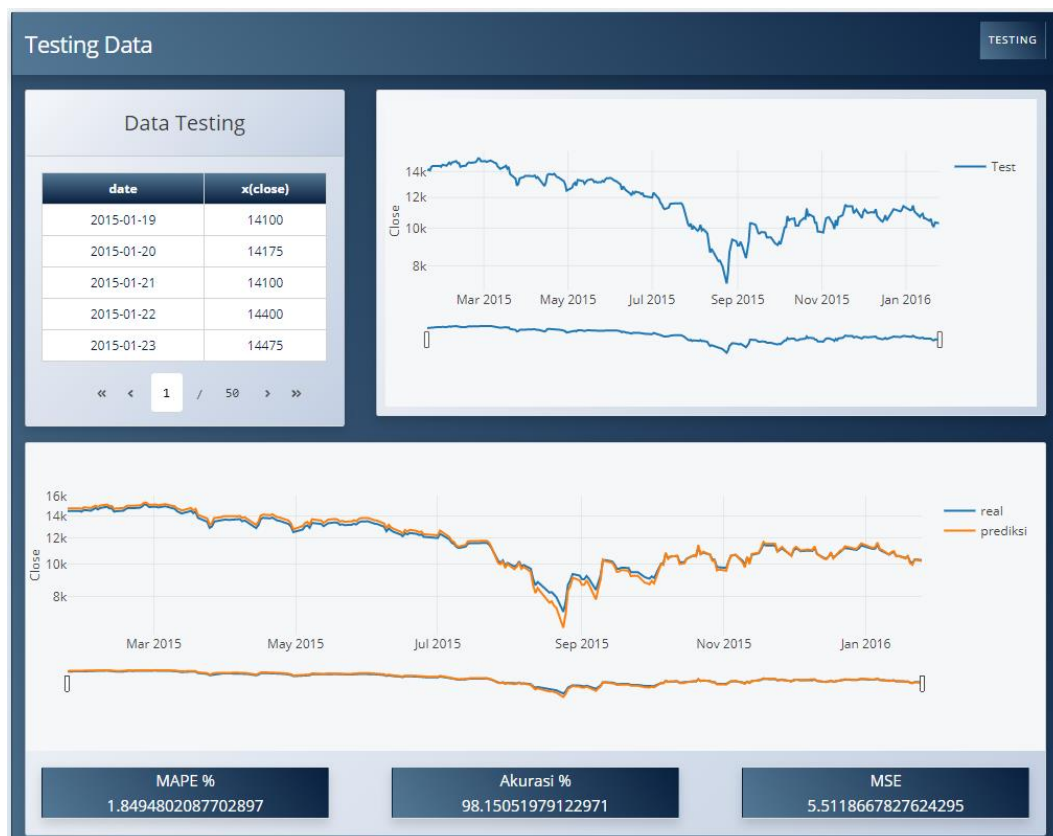
Table 45 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
100	0.1	474.09	19.06	80.94
100	0.05	1038.40	27.74	72.26
100	0.2	5.51	1.85	98.15
100	0.3	37.64	4.46	95.53
100	0.4	50.47	5.13	94.87

Table 46 Hasil terbaik Skenario 1

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
100	0.2	5.51	1.85	98.15

Akurasi error yang dihasilkan dengan menggunakan parameter yang terdapat pada tabel 4.5 mencapai 98.15%.



Gambar 4.28 Tampilan hasil testing data scenario 1

Skenario 2

Table 47 Data training dan data testing skenario 2

Ujicoba	Trainig	Testing
1	500 DATA	500 DATA

Penjelasan dari tabel 4.6 adalah total data yang dipakai adalah 1000 data saham, dari data tersebut dibagi lagi menjadi 500 data training dan 500 data testing. Dilakukan proses LSTM dengan melalui tahap sehingga memperoleh bobot baru Berikut hasil uji coba dari skenario 2.

Table 48 Perubahan Uji Coba dengan perubahan nilai epoch pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
5	0.2	15.03	3.40	96.60
10	0.2	25.32	4.70	95.30
25	0.2	3.60	1.88	98.12
50	0.2	2.28	1.26	98.74
100	0.2	1.73	1.10	98.90

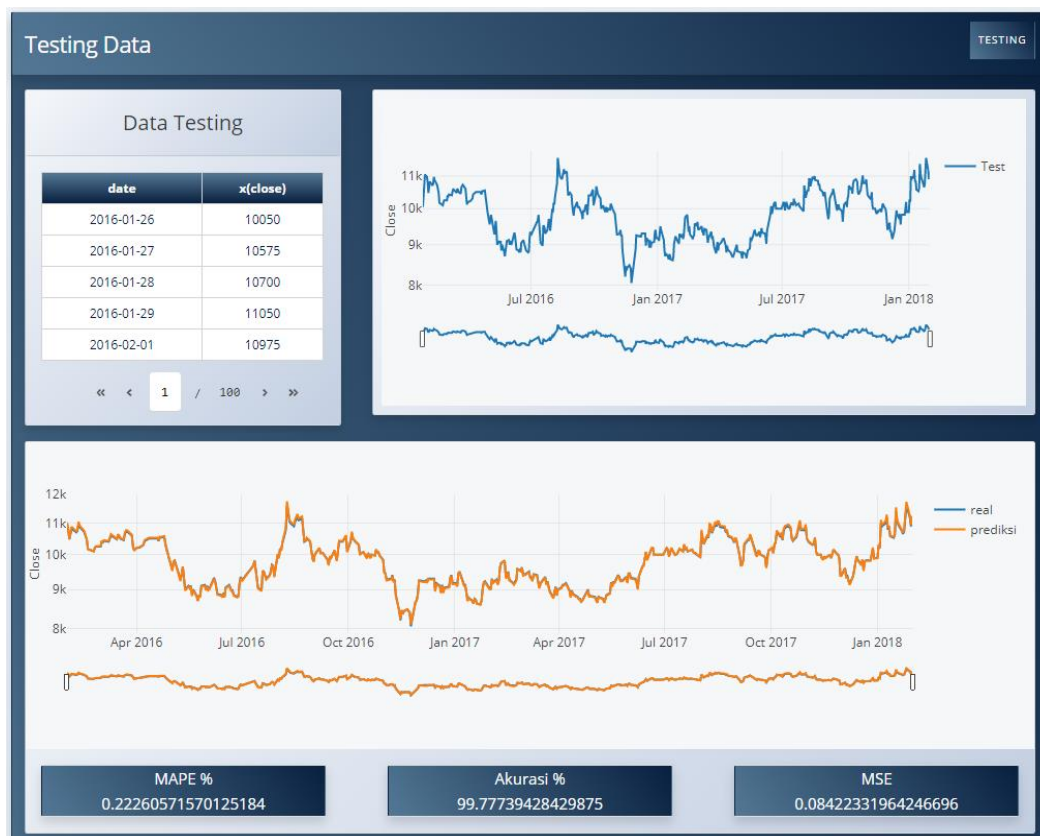
Table 49 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
100	0.05	2.24	1.21	98.79
100	0.1	2.46	1.28	98.71
100	0.2	1.73	1.10	98.90
100	0.3	0.08	0.22	99.78
100	0.4	0.29	0.47	99.53

Table 50 Hasil terbaik Skenario 1

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
100	0.3	0.08	0.22	99.78

Akurasi error yang dihasilkan dengan menggunakan parameter yang terdapat pada tabel 4.10 mencapai 99.68%.



Gambar 4.29. Tampilan hasil testing data skenario 2

Skenario 3

Table 51 Data training dan data testing skenario 3

Ujicoba	Trainig	Testing
1	1000 DATA	500 DATA

Penjelasan dari tabel 4.11 adalah total data yang dipakai adalah 1500 data saham, dari data tersebut dibagi lagi menjadi 1000 data training dan 500 data testing. Dilakukan proses LSTM dengan melalui tahap sehingga memperoleh bobot baru Berikut hasil uji coba dari skenario 3.

Table 52 Perubahan Uji Coba dengan perubahan nilai Epoch pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
5	0.2	144.51	9.91	90.09
10	0.2	59.00	6.27	93.73
25	0.2	6.00	1.87	98.13
50	0.2	0.51	0.38	99.62
100	0.2	0.87	0.61	99.40

Table 53 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
50	0.05	8.74	1.91	98.09
50	0.1	2.38	1.25	98.75
50	0.2	0.51	0.38	99.62
50	0.3	2.12	0.98	99.02
50	0.4	2.14	0.83	99.17

Table 54 Hasil terbaik Skenario 3

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
50	0.2	0.51	0.38	99.62

Akurasi error yang dihasilkan dengan menggunakan parameter yang terdapat pada tabel 4.14 mencapai 99.68%.



Gambar 4.130 Tampilan hasil testing data scenario 3

BAB V PENUTUP

5.1. Kesimpulan

Berdasarkan penelitian dan pengujian sistem yang telah dilakukan, maka disimpulkan bahwa:

1. Hasil uji coba yang dilakukan pada data harga saham menggunakan metode Neural Network algoritma LSTM menghasilkan keakuratan yang sangat bagus dengan 100 proses pelatihan dari 13 proses uji coba dengan berbagai nilai parameter learning rate yang berbeda - beda dan mendapatkan rata-rata akurasi 94,92 %
2. Dari hasil uji coba yang dilakukan diperoleh hasil prediksi dengan nilai parameter yang terbaik dengan rata-rata akurasi sebesar 99,18% dengan parameter yang berbeda-beda dari setiap skenario.
3. Dari tiga hasil skenario yang menghasilkan hasil terbaik adalah skenario ke tiga dengan parameter *epoch 50*, *learning rate 0.2* dan menghasilkan MSE 0.51 dan MAPE 0.38%, akurasinya 99.62%.

5.2 Saran

Saran yang diajukan untuk penelitian yang selanjutnya Penelitian ini digunakan untuk penelitian jangka pendek. Oleh karena itu untuk penelitian selanjutnya diharapkan agar dapat meramalkan harga saham dalam jangka panjang menggunakan metode metode lain.

DAFTAR PUSTAKA

- [1] A. Santoso and S. Hansun, “Prediksi IHSG dengan Backpropagation Neural Network,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 313–318, 2019, doi: 10.29207/resti.v3i2.887.
- [2] A. Wanto and A. P. Windarto, “Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan Dengan Menggunakan Metode Backpropagation,” *J. Penelit. Tek. Inform. Sink.*, vol. 2, no. 2, pp. 37–43, 2017.
- [3] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, “Comparison of Predictive Algorithms: Backpropagation, SVM, LSTM and Kalman Filter for Stock Market,” *Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019*, pp. 228–234, 2019, doi: 10.1109/AICAI.2019.8701258.
- [4] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [5] A. Arfan, “Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory,” vol. 3, 2019.
- [6] R. Maulana and D. Kumalasari, “Analisis Dan Perbandingan Algoritma Data Mining Dalam Prediksi Harga Saham GGRM,” *J. Inform. Kaputama*, vol. 3, no. 1, pp. 22–28, 2019.
- [7] L. Wiranda, M. Sadikin, J. T. Informatika, and F. I. Komputer, “Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt . Metiska Farma,” vol. 8, pp. 184–196, 2019.
- [8] P. J. Werbos, “Backpropagation Through Time: What It Does and How to Do It,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990, doi: 10.1109/5.58337.
- [9] S. Øyen, “Forecasting Multivariate Time Series Data Using Neural Networks,” *Nor. Univ. Sci. Technol. Dep. Eng. Cybern.*, no. June, 2018, doi: 10.1080/13540602.2012.629837.
- [10] C. Paper, W. Setiawan, and U. Trunojoyo, “Feedforward Network Dengan,” no. October, 2016, doi: 10.13140/2.1.3467.5525.
- [11] J. Lusikooy, N. Nainggolan, and J. Titaley, “Prediksi Harga Tutup Saham PT.

- Garuda Indonesia,Tbk Menggunakan Metode ARIMA,” *J. MIPA*, vol. 6, no. 1, p. 74, 2017, doi: 10.35799/jm.6.1.2017.16174.
- [12] E. Riyanto, “Peramalan Harga Saham Menggunakan Jaringan Syaraf Tiruan Secara Supervised Learning Dengan Algoritma Backpropagation,” *J. Inform. Upgris*, vol. 3, no. 2, pp. 137–142, 2017, doi: 10.26877/jiu.v3i2.1899.
- [13] A. Jk, D. Algoritma, S. V. M. Model, Y. Ramdhani, and A. Mubarak, “Analisis Time Series Prediksi Penutupan Harga Saham,” vol. 1, no. 1, pp. 77–82, 2019.
- [14] U. Islam, N. Imam, and B. Padang, “Forecasting Saham Syariah Dengan, LSTM ” 2019.
- [15] Mallya, A. (2017, January 24).
<http://arunmallya.github.io/writeups/nn/lstm/index.html#/>. (A. Mallya, Ed.)
 Retrieved from <http://arunmallya.github.io>:
<http://arunmallya.github.io/writeups/nn/lstm/index.html#/>

6. Lampiran Hasil Uji Coba

6.1 Hasil Uji Coba

Table 55 Skenario Uji Coba

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

Skenario 1

Table 56 Data training dan data testing skenario 1

Ujicoba	Trainig	Testing
1	250 DATA	250 DATA

Table 57 Perubahan Uji Coba dengan perubahan nilai epoch pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
25	0.2	978.08	27.28	72.72
50	0.2	411.52	17.89	82.11
100	0.2	5.51	1.85	98.15
200	0.2	169.28	10.05	89.95
400	0.2	261.98	12.50	87.5

Table 58 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
100	0.05	1038.40	27.74	72.26
100	0.1	474.09	19.06	80.94
100	0.2	5.51	1.85	98.15
100	0.3	37.64	4.46	95.53
100	0.4	50.47	5.13	94.87

Table 59 Hasil terbaik Skenario 1

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
100	0.2	5.51	1.85	98.15



Gambar 6.31 Tampilan hasil testing data scenario 1 terbaik

Skenario 2

Table 60 Data training dan data testing skenario 2

Ujicoba	Trainig	Testing
1	500 DATA	500 DATA

Table 61 Perubahan Uji Coba dengan perubahan nilai epoch pada data

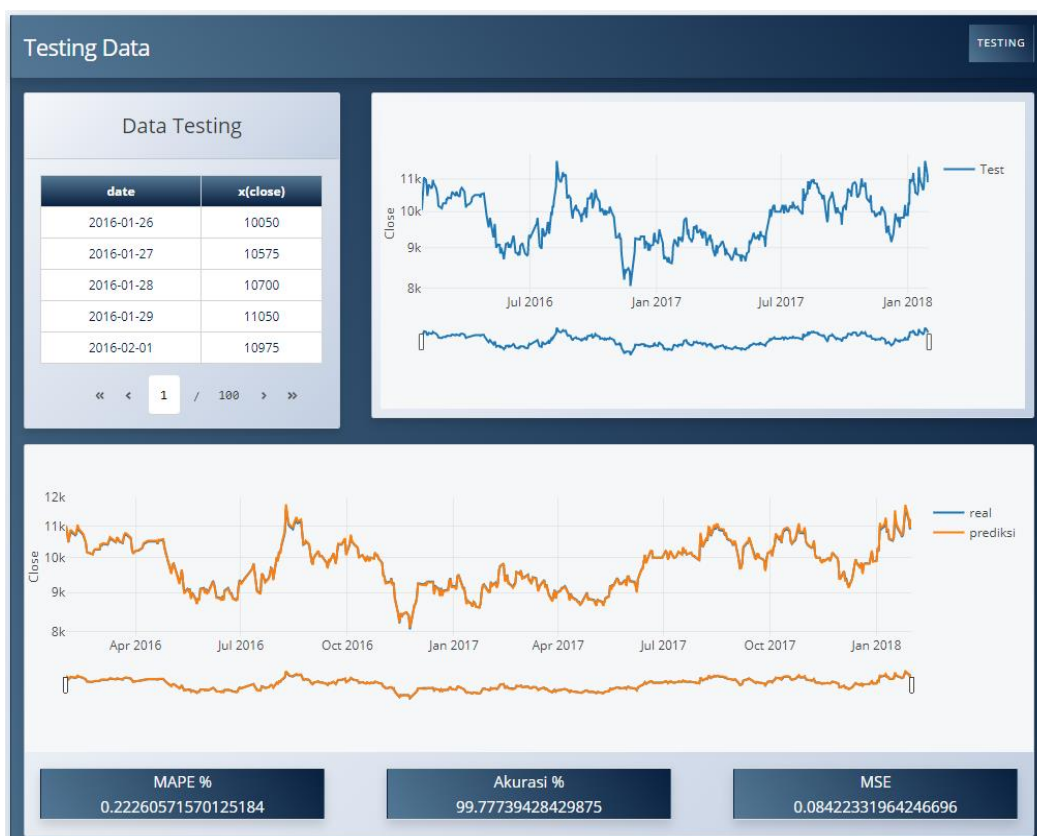
Epoch	Learning Rate	MSE	MAPE %	Akurasi %
25	0.2	3.60	1.88	98.12
50	0.2	2.28	1.26	98.74
100	0.2	1.73	1.10	98.90
200	0.2	0.50	0.60	99.40
400	0.2	0.11	0.30	99.72

Table 62 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
400	0.05	1.165	0.90	99.10
400	0.1	0.82	0.74	99.26
400	0.2	0.11	0.30	99.72
400	0.3	0.44	0.49	99.51
400	0.4	0.63	0.60	99.40

Table 63 Hasil terbaik Skenario 1

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
400	0.2	0.11	0.30	99.72



Gambar 6.32. Tampilan hasil testing data scenario 2 terbaik

Skenario 3

Table 64 Data training dan data testing skenario 3

Ujicoba	Trainig	Testing
1	1000 DATA	500 DATA

Table 65 Perubahan Uji Coba dengan perubahan nilai Epoch pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
25	0.2	6.00	1.87	98.13
50	0.2	0.51	0.38	99.62
100	0.2	0.87	0.61	99.40
200	0.2	1.81	0.88	99.16
400	0.2	1.94	0.87	99.13

Table 66 Perubahan Uji Coba dengan perubahan nilai Learning rate pada data

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
50	0.05	8.74	1.91	98.09
50	0.1	2.38	1.25	98.75
50	0.2	0.51	0.38	99.62
50	0.3	2.12	0.98	99.02
50	0.4	2.14	0.83	99.17

Table 67 Hasil terbaik Skenario 3

Epoch	Learning Rate	MSE	MAPE %	Akurasi %
50	0.2	0.51	0.38	99.62



Gambar 6.3 Tampilan hasil testing data scenario 3 terbaik

