

SKRIPSI

**Analisis Kinerja *Long Short-Term Memory*,
Peephole Connection LSTM dan *Facebook's Prophet*
dalam Memprediksi Pergerakan Harga Saham
PT Telekomunikasi Indonesia Tbk**

Disusun dan diajukan oleh :

FADHILLAH PUTRI TAHA

H071171301



**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
JULI 2021**

**Analisis Kinerja *Long Short-Term Memory*,
Peephole Connection LSTM dan *Facebook's Prophet*
dalam Memprediksi Pergerakan Harga Saham
PT Telekomunikasi Indonesia Tbk**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer pada Program Studi Sistem Informasi
Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Hasanuddin**

FADHILLAH PUTRI TAHA

H071171301

**PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN**

MAKASSAR

JULI 2021

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Fadhillah Putri Taha
NIM : H071171301
Program Studi : Sistem Informasi
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**Analisis Kinerja Long Short-Term Memory, Peephole Connection LSTM dan
Facebook's Prophet dalam Memprediksi Pergerakan Harga Saham
PT Telekomunikasi Indonesia Tbk**

adalah karya tulisan saya sendiri dan bukan merupakan pengambil alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 30 Juli 2021

Yang menyatakan,



Fadhillah Putri Taha
NIM. H071171301

**Analisis Kinerja *Long Short-Term Memory*,
Peephole Connection LSTM dan *Facebook's Prophet*
dalam Memprediksi Pergerakan Harga Saham
PT Telekomunikasi Indonesia Tbk**

Disusun dan diajukan oleh

FADHILLAH PUTRI TAHA

H071171301

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin pada tanggal 30 Juli 2021 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama



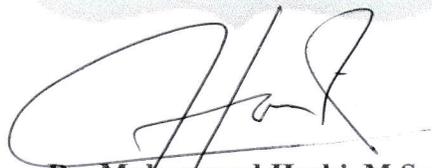
Dr. Hendra, S.Si., M.Kom.
NIP. 19760102 200212 1 001

Pembimbing Pertama



Supri Bin Hj. Amir, S.Si., M.Eng.
NIP. 19880504 201903 1 012

Ketua Program Studi,



Dr. Muhammad Hasbi, M.Sc.
NIP. 19630720 198903 1 003



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Fadhillah Putri Taha
NIM : H071171301
Program Studi : Sistem Informasi
Judul Skripsi : Analisis Kinerja *Long Short-Term Memory*, *Peephole Connection LSTM* dan *Facebook's Prophet* dalam Memprediksi Pergerakan Harga Saham PT Telekomunikasi Indonesia Tbk

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

Ketua : Dr. Hendra, S.Si., M.Kom.

Sekretaris : Supri Bin Hj. Amir, S.Si., M.Eng.

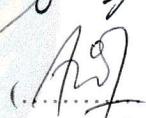
Anggota : A. Muh. Amil Siddik, S.Si., M.Si.

Anggota : Edy Saputra, S.Si., M.Si.

Tanda tangan


.....)


.....)


.....)


.....)

Ditetapkan di : Makassar
Tanggal : 30 Juli 2021



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan pendidikan jenjang Strata 1 pada Program Studi Sistem Informasi, Universitas Hasanuddin. Tugas akhir ini disusun guna memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer (S.Kom.).

Penulis menyadari bahwa telah mendapatkan banyak bantuan serta bimbingan dari berbagai pihak, dari masa perkuliahan hingga pada tahap akhir penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini, dengan segala kerendahan hati penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Pemerintah melalui Direktorat Jendral Pembelajaran dan Kemahasiswaan (Ditjen Belmawa), Kementrian Riset, Teknologi dan Perguruan Tinggi (Kemenristekdikti) yang telah membantu pembiayaan selama perkuliahan melalui program **Basiswa Bidikmisi**. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubulu, MA.** beserta jajarannya, Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Eng. Amiruddin, S.Si.** beserta jajarannya, dan Ketua Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Muhammad Hasbi, M.Sc.** beserta seluruh jajarannya. Tak lupa pula Bapak Wakil Dekan Bidang Kemahasiswaan **Dr. Andi Ilham Latunra, M.Sc.** yang telah memberikan kesan dan banyak cerita selama penulis menjadi pengurus **BE Himatika FMIPA Unhas**.
2. Orang tua penulis, ayahanda **Achmad Taha** dan ibunda **Amiluddinul Qoyimah** yang tidak henti-hentinya menyelipkan nama anaknya di dalam doa kebaikan, adik-adik serta kakek dan nenek, tehusus kepada tante tercinta **Adnin Taha** yang tidak pernah lelah membantu dan memotivasi penulis hingga berada di titik yang luar biasa ini.
3. Kakanda **Muh. Alfajar** sebagai penyejuk pikiran di setiap kegundahan masa-masa sulit, juga di hari-hari senang penulis yang semoga masih dapat selalu saling menyemangati dan membantu satu sama lain hingga nanti.

4. Bapak **Dr. Hendra, S.Si., M.Kom.** dan Bapak **Supri Bin Hj. Amir, S.Si., M.Eng.** atas kesediaan, kesabaran, dan waktu yang telah diluangkan untuk membimbing penulis selama proses penyusunan tugas akhir. Juga Bapak
5. **A. Muh. Amil Siddik, S.Si. M.Si.** dan bapak **Edy Saputra, S.Si., M.Si.** atas waktu dan kesediannya sebagai penguji pada tahap penyusunan tugas akhir.
6. Dosen Departemen Matematika, dan terkhusus kepada Bapak dan Ibu dosen Program Studi Sistem Informasi, Fakultas MIPA, Universitas Hasanuddin atas semua ilmu yang diberikan kepada penulis selama menempuh Pendidikan.
7. Seluruh warga **KM FMIPA** dan **Himatika FMIPA Unhas** sebagai keluarga kedua penulis yang memberikan kenangan dan pelajaran hidup yang tidak terlupakan. Khususnya, seluruh saudara-saudara **DISKRIT 2017**, ketua angkatan **Ihzanul Wajdi**, saudara **Nur Fathir Supardi** yang senantiasa membimbing penulis dalam menyelesaikan tugas perkuliahan, **Ali Ahmad Ali Winandar, Aris Akhyar Abdillah**, dan **Muh. Rifky Wahyudi Bonda** serta saudari-saudari **Komath-Kamith (Tasya Wiraz Miftahuljannah** yang senantiasa membimbing penulis dalam menyelesaikan tugas perkuliahan, **Krisdayanti** selaku koordinator Divisi Kesekretariatan dan ibu angkatan, **Indi RiaAl-Kadry, Alif Miftahul Jannah, Khandy Dilsab Apriliani, Nur Khairunisa** sebagai seorang yang selalu ada, **Eka Kurnia**, dan **Nurfadila Firdani Salam**) terima kasih atas kebersamaan, loyalitas, semangat dan doa yang diberikan kepada penulis.
8. Teman-teman **Gepeng (Nur Khairunisa** sebagai pengingat tugas dan teman pulang dari kampus, **Eka Kurnia** sebagai motivator hidup, **Eka Fitriani** sebagai teman yang berjiwa keibuan bagiku, **Nurfadila Firdani Salam, Geby Nionsi, Azzahra Mubarikah**, dan **Mir Ataini Aprilia**), atas loyalitasnya sejak menjadi mahasiswa baru Ilmu Komputer hingga berganti nama menjadi Sistem Informasi, terima kasih sudah menjadi teman hidup penulis selama 4 tahun menjalani perkuliahan, berbagi suka dan duka selama menjadi mahasiswi hingga bukan mahasiswi lagi. Tak lupa pula teman-teman **NGINX (Muthia Amanah Arum dan Siti Rabiatul Adawiyah)**, sebagai *team-mate*

- dalam beberapa proyek yang memiliki kekompakan sangat luar biasa dan dapat saling menunjang efektivitas serta efisiensi kinerja tim di sela-sela kesibukan.
9. Kakanda **Ilham** selaku mantan Koordinator Laboratorium Rekayasa Perangkat Lunak FMIPA Unhas yang sangat banyak membantu penulis dalam membuat tugas dari semester 1 hingga semester akhir. Dan teman-teman **Ilmu Komputer 2017** (berubah menjadi: Sistem Informasi awal tahun 2021) yang memberi dukungan dan kerjasamanya selama kurang lebih 4 tahun di Universitas Hasanuddin. Terkhusus kepada **Muhammad Fitrah, Denny Pratama Hardiono, Farhan Ramdhani, Aris Akhyar Abdillah**, sebagai teman pengajar yang tabah dan tekun selama perkuliahan, tak lupa pula saudara **Khawaritzmi Abdallah Ahmad**.
 10. Kakanda karyawan *remote* AWS dan temannya yang semoga disegerakan dapat menjadi karyawan *remote* *Google*, yang telah meluangkan banyak waktu berharganya, juga sangat luar biasa dalam membantu penulis dalam mngimplementasikan beberapa metode *forecasting* pada tugas akhir ini. Sebagai kakanda yang telah memberikan banyak kesan tak terlupakan sejak percakapan pertama hingga akhir, sehingga penulis sangat berharap masih dapat menjalin komunikasi dan dapat bertemu langsung di lain kesempatan. Perkenalan singkat ini merupakan suatu kehormatan dan kebanggaan tersendiri bagi penulis yang dapat menjalin silaturahmi dengan kakanda di waktu sibuk.
 11. **Facebook, StackOverflow, Google, GeeksforGeeks, GitHub, Machine Learning Mastery**, dan *website-website* lainnya sebagai sumber pembelajaran yang menunjang penyelesaian tugas-tugas perkuliahan serta penyusunan tugas akhir ini. Tak lupa pula jurnal-jurnal yang berkaitan dengan penelitian ini sebagai pedoman penyusunan tugas akhir.

Makassar, 30 Juli 2021


Fadhillah Putri Taha

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Fadhillah Putri Taha
NIM : H071171301
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

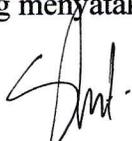
Analisis Kinerja Long Short-Term Memory, Peephole Connection LSTM dan Facebook's Prophet dalam Memprediksi Pergerakan Harga Saham PT Telekomunikasi Indonesia Tbk

berserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 30 Juli 2021

Yang menyatakan



(Fadhillah Putri Taha)

ABSTRAK

Peramalan pasar saham memiliki kompleksitas yang tinggi, sehingga telah menjadi objek studi selama bertahun-tahun karena merupakan tugas yang sulit. Peningkatan penggunaan algoritma *machine learning* menyebabkan evolusi teknik dan metode berbeda dalam ilmu pembelajaran mesin dari masa ke masa dalam membuat peramalan karena pergerakan harga di pasar saham sangat dinamis, sehingga memerlukan pemodelan data untuk melakukan prediksi harga saham agar mendapatkan harga dengan tingkat kesalahan yang rendah. Di dalam penelitian ini membahas 3 pemodelan data, yaitu *Vanilla Long Short-Term Memory* (LSTM), *Peephole Connection LSTM*, dan *Facebook's Prophet* untuk meramalkan pergerakan harga saham yang kemudian model dievaluasi dengan menghitung tingkat kesalahan/*error* peramalan menggunakan *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE). Hasil penelitian menunjukkan bahwa *Peephole Connection LSTM* memiliki performa yang lebih baik dalam meramalkan pergerakan harga saham, dibandingkan *Vanilla LSTM* dan *Facebook's Prophet* berdasarkan nilai *error testing* untuk *Vanilla LSTM* dengan MAE senilai 223.219 dan RMSE senilai 236.245, model *Peephole Connection* dengan MAE senilai 113.166 dan RMSE senilai 121.475, dan *Prophet* dengan MAE senilai 129.828 dan RMSE senilai 180.827.

Kata kunci: Peramalan, *Recurrent Neural Network*, *Long Short-Term Memory*, *Peephole Connection LSTM*, *Prophet Facebook*, Harga Saham

ABSTRACT

Stock market forecasting has a high degree of complexity, so it has been the object of study for many years as it is a difficult task. The increasing use of machine learning algorithms has led to the evolution of different techniques and methods in machine learning from time to time in making forecasts because price movements in the stock market are very dynamic, thus requiring data modeling to predict stock prices in order to get prices with a low error rate. This study discusses 3 data modeling, there is Vanilla Long Short-Term Memory (LSTM), Peephole Connection LSTM, and Facebook's Prophet to predict stock price movements which then the model is evaluated by calculating the level of forecasting error / error using Mean Absolute Error (MAE). and Root Mean Squared Error (RMSE). The results show that the Peephole Connection LSTM has a better performance in predicting stock price movements, compared to Vanilla LSTM and Facebook's Prophet based on error testing values for Vanilla LSTM with MAE worth 223.219 and RMSE worth 236.245, Peephole Connection model with MAE worth 113166 and RMSE worth 121.475, and Prophet with MAE worth 129.828 and RMSE worth 180.827.

Keywords: Forecasting, Recurrent Neural Network, Long Short-Term Memory, Peephole Connection LSTM, Prophet Facebook, Stock Price.

DAFTAR ISI

PERNYATAAN KEASLIAN.....	iii
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR	ix
ABSTRAK	x
ABSTRACT.....	xi
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	4
1.5. Batasan Masalah.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Penelitian Terdahulu	5
2.2. Kerangka Teori.....	6
2.2.1. <i>Forecasting</i>	6
2.2.2. <i>Stock / Saham</i>	7
2.2.3. <i>Time Series Analysis</i>	8
2.2.4. <i>Recurrent Neural Network</i>	8
2.2.5. <i>Long Short-Term Memory</i>	9
2.2.5.1. <i>Forget Gate</i>	11
2.2.5.2. <i>Input Gate</i>	12
2.2.5.3. <i>Memory Update Gate</i>	13
2.2.5.4. <i>Output Gate</i>	14
2.2.6. <i>Peephole Connection LSTM</i>	15
2.2.7. <i>Facebook's Prophet</i>	17

2.2.7.1. <i>Trend Model</i>	19
2.2.7.2. <i>Seasonality</i>	20
2.2.7.3. <i>Holidays dan Events</i>	21
2.2.8. <i>Mean Absolute Error (MAE)</i>	23
2.2.9. <i>Root Mean Square Error (RMSE)</i>	23
2.2.10. <i>Normalisasi Data</i>	24
BAB III METODE PENELITIAN	25
3.1. <i>Dataset</i>	25
3.2. <i>Instrumen Penelitian</i>	25
3.3. <i>Tahapan Penelitian</i>	25
3.4. <i>Metode Penelitian</i>	26
3.4.1. <i>Proses Long Short-Term Memory (LSTM)</i>	27
3.4.2. <i>Proses Prophet Facebook</i>	28
BAB IV HASIL DAN PEMBAHASAN	30
4.1. <i>Deskripsi Dataset</i>	30
4.2. <i>Alur Kerja Implementasi</i>	30
4.2.1. <i>Data Preparation</i>	30
4.2.2. <i>Modelling</i>	31
4.2.2.1 <i>Long Short-Term Memory</i>	31
4.2.2.2 <i>Facebook's Prophet</i>	34
4.2.3. <i>Hasil Implementasi</i>	38
4.3. <i>Analisis Kinerja</i>	39
BAB V KESIMPULAN DAN SARAN	42
5.1. <i>Kesimpulan</i>	42
5.2. <i>Saran</i>	42
Daftar Pustaka	43
LAMPIRAN	47

DAFTAR TABEL

Tabel 3.1. Sampel data tiga hari pertama dan 3 hari terakhir	25
Tabel 4.1. Data Harga yang Dinormalisasi	31
Tabel 4.2. Segmentasi Data <i>Train</i>	32
Tabel 4.3. Segmentasi Data <i>Test</i>	32
Tabel 4.4. Model <i>Sequential Vanilla LSTM</i>	33
Tabel 4.5. Model <i>Sequential Peephole Connection LSTM</i>	34

DAFTAR GAMBAR

Gambar 2.1. Arsitektur LSTM.....	10
Gambar 2.2. Mekanisme Konteks Memori LSTM	11
Gambar 2.3. <i>Forget Gate</i>	12
Gambar 2.4. <i>Input Gate</i>	13
Gambar 2.5. <i>Memory Update Gate</i>	14
Gambar 2.6. <i>Output Gate</i>	14
Gambar 2.7. <i>Peephole Connection LSTM</i>	16
Gambar 3.1. Diagram Alur Penelitian.....	26
Gambar 3.2. Diagram Alur Pemrosesan Model LSTM	27
Gambar 3.3. Diagram Alur Pemrosesan Model LSTM	28
Gambar 4.1. Pergerakan Harga Penutupan Saham TLKM	30
Gambar 4.2. Segmentasi Keseluruhan Data.....	32
Gambar 4.3. Komponen <i>Tren Training Prophet</i>	35
Gambar 4.4. Komponen <i>Holidays Training Prophet</i>	35
Gambar 4.5. Komponen <i>Weekly Seasonality Training Prophet</i>	36
Gambar 4.6. Komponen <i>Yearly Seasonality Training Prophet</i>	36
Gambar 4.7. Komponen <i>Tren Testing Prophet</i>	37
Gambar 4.8. Komponen <i>Holidays terhadap Data Test Prophet</i>	37
Gambar 4.9. Perbandingan <i>Training Model</i>	38
Gambar 4.10. Perbandingan <i>Testing Model</i>	39
Gambar 4.11. <i>Mean Absolute Error</i>	39
Gambar 4.12. <i>Root Mean Squared Error</i>	40
Gambar 4.13. <i>Barplot Durasi Training LSTM dan Prophet</i>	41
Gambar 4.14. Kurva Laju Penurunan Per- <i>Epoch</i>	41

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pasar modal merupakan pasar untuk berbagai instrumen keuangan jangka panjang yang dapat diperjualbelikan, seperti obligasi (surat utang), saham (ekuiti), reksa dana, dan berbagai instrumen lainnya. Transaksi jual beli harga saham merupakan salah satu bentuk investasi favorit yang dilakukan banyak orang, karena mendapatkan keuntungan yang menjanjikan sesuai dengan tingkatan resiko yang diberikan. Ketelitian serta melakukan penilaian terhadap suatu emiten/perusahaan pemilik lembar saham merupakan tindakan penting dan berguna dalam mengurangi resiko kerugian bagi investor dalam berinvestasi. Harga saham dapat mengalami perubahan dalam beberapa periode, misalnya periode setiap hari, minggu, bulan, dan tahun. Terdapat berbagai faktor yang dapat mempengaruhi pergerakan harga saham, di antaranya adalah sentimen atau berita perusahaan, berapa banyak perbandingan transaksi jual dan beli, dan lain sebagainya.

Sebagai investor harus selalu melakukan analisis terhadap suatu saham sebelum membeli, di antaranya analisis fundamental dan analisis teknikal. Teknik-teknik statistika berperan dalam peramalan pergerakan harga saham, seperti stokastik, metode fibonacci, *moving average* (MA), EMA, dan lain sebagainya. Hasil peramalan teknik statistika tersebut diperlukan dalam proses pengambilan keputusan.

Peramalan pasar saham memiliki kompleksitas yang tinggi, sehingga telah menjadi objek studi selama bertahun-tahun karena merupakan tugas yang sulit. Peningkatan penggunaan algoritma *machine learning* menyebabkan evolusi teknik dan metode berbeda dalam ilmu pembelajaran mesin dari masa ke masa dalam membuat peramalan. Artikel-artikel sebelumnya telah menunjukkan bahwa terdapat banyak faktor seperti variabel, parameter, indikator, bahasa pemrograman, perangkat lunak, dan jaringan model yang dapat sangat mempengaruhi hasil peramalan harga. Berbagai jenis *neural network* telah diterapkan dalam

meramalkan harga dan operasi lainnya. Misalnya, *artificial neural network* (ANN), *convolutional neural network* (CNN), *autoencoders* (AE), *boltzmann machine* (BM), *extreme learning machine* (ELM), dan *autoregressive integrated moving average* (ARIMA) telah digunakan untuk peramalan harga. Tetapi dalam penelitian ini menggunakan metode *Long Short-Term Memory Model* (LSTM), *Peephole Connection LSTM*, dan *Facebook's Prophet*.

Ada beberapa alasan utama mengapa *deep neural network* berfungsi dengan baik. Pada skenario *big data*, *deep neural network* memiliki kemampuan generalisasi yang lebih baik daripada *neural network* biasa. Alasan lain untuk keberhasilan jaringan neural dalam adalah struktur hierarki dari beberapa lapisan jaringan yang dapat dianggap sebagai *pipeline* pemrosesan, setiap lapisan memproses sebagian tugas sebelum meneruskan tugas ke lapisan berikutnya, hingga lapisan terakhir menghasilkan keluaran.

Long short term memory adalah salah satu arsitektur RNN yang paling sukses. LSTM memperkenalkan sel memori, sebuah unit komputasi yang menggantikan *neuron* buatan tradisional di lapisan tersembunyi jaringan. Dengan sel memori ini, jaringan dapat secara efektif mengasosiasikan memori dan *input* jarak jauh dalam waktu, sehingga cocok untuk memahami struktur data secara dinamis dari waktu ke waktu dengan kapasitas peramalan yang tinggi (Roondiwala, 2017). Jaringan saraf dipengaruhi oleh otak manusia dan mengandung beberapa *neuron* yang membentuk jaringan yang mengesankan (Vateekul, 2016). RNN diibaratkan *lobus frontalis* yang merupakan bagian terbesar dari otak dan juga bertanggung jawab atas ingatan jangka pendek. Ingatan jangka pendek merupakan faktor yang sangat penting dalam meramalkan harga karena jaringan dapat mengingat nilai sebelumnya dan dapat menghitung nilai di masa depan berdasarkan akurasi nilai sebelumnya (Tawum, 2021).

Long short term memory semakin mengalami perkembangan, salah satunya adalah penambahan lubang intip (*Peephole Connection*). *Peephole Connection* memungkinkan gerbang-gerbang tidak hanya bergantung pada *hidden state* sebelumnya (h_{t-1}), tetapi juga pada keadaan internal/*cell state* sebelumnya (c_{t-1})

dan menambahkan nilai tambahan di setiap *gate*. Sehingga dengan menambahkan *Peephole Connection*, maka hasil *training* akan meningkat.

Facebook menjadikan *Prophet Facebook* sebagai *library open source* yang sebelumnya hanya dipakai secara internal. *Prophet* merupakan alat untuk menyelesaikan berbagai kasus dalam dunia bisnis. Dalam data *timeseries*, analis dapat meningkatkan ketepatan peramalan dengan mengubah berbagai parameter yang mudah diinterpretasikan. *Facebook* menggabungkan perkiraan otomatis dengan *analyst-in-the-loop forecasts* untuk kasus-kasus khusus, dimungkinkan untuk mencakup berbagai kasus penggunaan bisnis (Taylor, 2017).

Pokok permasalahan penelitian ini adalah membandingkan *error* model sistem yang dibuat dalam meramalkan pergerakan harga saham PT Telekomunikasi Indonesia (Persero) Tbk menggunakan 3 model jaringan, yaitu *Long Short-Term Memory Model*, *Peephole Connection LSTM*, dan *Facebook's Prophet*. Sebagai masukannya, menggunakan harga penutupan selama kurang lebih 16 tahun. Model diimplementasikan menggunakan bahasa pemrograman *Python*. Selanjutnya model dites menggunakan data. Sebagai evaluasi, *error* model dihitung menggunakan metode RMSE dan MAE.

1.2. Rumusan Masalah

Berikut beberapa poin rumusan masalah penelitian :

1. Bagaimana mengimplementasikan model LSTM, *Peephole Connection LSTM*, dan *Facebook's Prophet* dalam meramalkan pergerakan harga saham TLKM?
2. Bagaimana kinerja model LSTM, *Peephole Connection LSTM*, dan *Facebook's Prophet* dalam meramalkan harga saham TLKM?

1.3. Tujuan Penelitian

Berikut beberapa poin tujuan dilakukan penelitian :

1. Mengimplementasikan model LSTM, *Peephole Connection LSTM*, dan *Facebook's Prophet* dalam meramalkan harga saham TLKM;
2. Membandingkan performa kinerja model LSTM, *Peephole Connection LSTM*, dan *Facebook's Prophet* dalam meramalkan harga saham TLKM.

1.4. Manfaat Penelitian

Penelitian ini sebagai implementasi dari beberapa metode *forecasting* diharapkan dapat dijadikan pembandingan dengan penelitian lain dalam bidang *data science*, dalam hal ini adalah *forecasting* harga saham.

1.5. Batasan Masalah

Mengingat cakupan masalah cukup luas, maka terdapat beberapa batasan agar penelitian ini dapat terfokus. Berikut batasan masalah yang diberikan :

1. Penelitian ini menggunakan data primer yang dikumpulkan dari website Yahoo! Finance dengan kode TLKM.jk;
2. Data yang digunakan berupa data numerik (harga dalam Rupiah) dengan 1 variabel, yaitu harga *close*;
3. Algoritma yang digunakan adalah LSTM, *Peephole Connection LSTM*, dan *Facebook's Prophet*;
4. Menggunakan bahasa pemrograman *Python*;
5. *Output* yang dihasilkan program, yaitu Nilai *error* MAE dan RMSE;

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Beberapa penelitian telah mengaplikasikan berbagai teknik dalam membuat peramalan harga. Tawum dkk., pada tahun 2020 membandingkan LSTM dan ARIMA dalam simulasi dan peramalan variasi harga batu gamping dengan akurasi SMAPE senilai 91.8% dan 95.7%. Ravikumar dan Saraf pada tahun 2020 membuat peramalan harga menggunakan algoritma regresi dan klasifikasi, akurasi terendah menggunakan *simple linear regression* senilai 81.52% dan akurasi tertinggi menggunakan *random forest regression* senilai 99.57%. Guangyu dan Liangxi pada tahun 2019 meramalkan beberapa nilai dalam satu model. Model jaringannya dapat meramalkan harga pembukaan, harga terendah, dan harga tertinggi suatu saham secara bersamaan. Metode yang digunakan adalah *deep recurrent neural network* (DRNN) dan LSTM dan menunjukkan bahwa akurasi model DRNN memiliki akurasi peramalan senilai 95% sedangkan LSTM hanya mendapatkan rata-rata akurasi senilai 0.78%. David dkk., pada tahun 2017 meramalkan pergerakan menggunakan LSTM *neural network* mendapatkan rata-rata akurasi peramalan senilai 55.9%. Yanhui pada tahun 2017 meramalkan harga minyak mentah dan mendapatkan kesimpulan bahwa kinerja optimal dicapai dengan menggunakan model RW-DBN ketika UMK terkecil di antara semua model. Secara umum, UMK model berbasis DBN lebih rendah daripada model berbasis LSTM. Hal ini menunjukkan kekuatan peramalan yang unggul dari model DBN jika dibandingkan dengan model LSTM, di pasar minyak mentah WTI. Ganesh Anirudh Panthula pada tahun 2018 meramalkan menggunakan LSTM *neural network* berdasarkan data histori harga penutupan saham S&P 500 dari Januari 2000 – Agustus 2016 dengan hasil RMSE data latih senilai 0.04, dan MSE senilai 0.00198 dan RMSE data tes senilai 0.13 dan MSE senilai 0.01802.

Lei pada tahun 2020 meramalkan data deret waktu menggunakan *Deep Peephole LSTM* (DPLSTM) dan menghasilkan performa yang terbaik

dibandingkan model ARIMA, *Single RNN*, *Deep RNN*, dan DGRU dengan DPLSTM mendapatkan nilai *error* MAE minimum senilai 0.008, RMSE minimum 0.013, dan RMSPE senilai 3.90. Siyuan dkk., pada tahun 2018 meramalkan transaksi saham menggunakan *Peephole Connection LSTM* mendapatkan hasil akurasi tidak begitu tinggi dengan hanya mendapatkan 72%.

Madhuri, dkk pada tahun 2020 memprediksi menggunakan *Prophet* pada data *Gold/Silver/Crude Oil Nifty 50 shares*. Dan menyimpulkan bahwa *Prophet* dirancang khusus untuk menganalisis data deret waktu pada pengamatan harian yang akan menampilkan pola pada berbagai jenis skala waktu. Ini dapat menjadi alat yang baik yang dapat digunakan dalam mengambil keputusan bagi pelanggan untuk memutuskan apakah akan membeli atau menjual saham tertentu karena *Prophet* sangat mudah digunakan meskipun bukan seorang statistikawan. Wen-Xiang dkk., pada tahun 2019 menggunakan *Prophet Facebook* dan dapat meramalkan tren harga saham MSF selama 60 hari dan hasil penelitian menunjukkan bahwa nilai peramalan dari saraf tipe transfer terbalik LSTM terintegrasi hampir sama dengan nilai aktual dua kurva, yang menunjukkan bahwa saraf tipe transfer balik LSTM dapat secara akurat meramalkan MSF dengan RMSE senilai 0.04464.

2.2. Kerangka Teori

2.2.1. Forecasting

Forecasting atau peramalan didefinisikan sebagai alat atau teknik meramalkan suatu nilai pada masa yang akan datang dengan memperhatikan data atau informasi masa lalu maupun pada saat ini. Peramalan dapat menjadi dasar bagi perencanaan jangka panjang suatu perusahaan/instansi tertentu (Luthfianto, 2011).

Di dalam *website* Jurnal Entrepreneur mengutip bahwa salah satu jenis peramalan berdasarkan data yang disusun dibagi menjadi dua jenis, yaitu:

1. Peramalan kualitatif, yang didasarkan atas data kualitatif pada masa lalu. Hasil peramalan yang dibuat sangat tergantung pada orang yang menyusunnya.

Biasanya perkiraan ini didasarkan atas hasil penyelidikan, seperti pendapat salesman, pendapat sales manajer pendapat para ahli, dan survey konsumen.

2. Peramalan kuantitatif, yaitu perkiraan yang didasarkan atas data penjualan pada masa lalu. Hasil perkiraan yang dibuat sangat tergantung pada metode yang digunakan dalam perkiraan tersebut. Penggunaan metode yang berbeda akan diperoleh hasil yang berbeda pula (Adisaputro, 2000).

2.2.2. Stock / Saham

Saham merupakan salah satu yang banyak diminati oleh investor dibursa efek (*stock exchange*). Saham dapat didefinisikan sebagai tanda atau pemilikan seseorang atau badan dalam suatu perusahaan atau perseroan terbatas. Saham berwujud selembar kertas yang menerangkan bahwa pemilik kertas adalah pemilik perusahaan yang menerbitkan surat berharga tersebut. Porsi kepemilikan ditentukan oleh seberapa besar penyertaan yang ditanamkan di perusahaan tersebut.

Saham dikenal dengan karakteristik imbal hasil tinggi, resiko tinggi. Artinya, saham merupakan surat berharga yang memberikan peluang keuntungan dan potensi resiko yang tinggi. Saham memungkinkan investor untuk mendapatkan imbalan hasil atau *capital gain* yang besar dalam waktu singkat. Namun seiring berfluktuasinya harga saham, maka saham juga dapat membuat investor mengalami kerugian besar dalam waktu singkat. Pembentukan harga saham terjadi karena adanya permintaan dan penawaran atas saham tersebut. Dengan kata lain, harga saham terbentuk atas permintaan dan penawaran saham (Gemilang, 2017).

Harga *open* adalah harga pembukaan di hari tersebut, harga *low* adalah harga terendah pada hari tersebut, harga *high* adalah harga tertinggi di hari tersebut, harga *close* adalah harga penutupan di hari tersebut, *Adjusted close* adalah harga penutupan setelah penyesuaian untuk semua pembagian dan pembagian dividen yang berlaku. Data disesuaikan menggunakan penganda pembagian dan dividen yang sesuai, mengikuti standar *Center for Research in Security Prices (CRSP)* (Yahoo!, 2021). Volume adalah jumlah aset/saham yang telah diperdagangkan dalam periode waktu tertentu. Misal dalam satu hari terdapat aksi jual-beli sebanyak 5 juta lembar saham. Maka itulah besar volume pada hari itu.

2.2.3. Time Series Analysis

Bagas (2019) mengutip dalam jurnal berjudul “*Perbandingan Prediksi Harga Saham dengan Model ARIMA dan Artificial Neural Network*”, bahwa *time series* adalah rangkaian data yang berupa nilai pengamatan yang diukur selama kurun waktu tertentu, berdasarkan waktu dengan interval yang sama (Spiegel, 2007), dalam hal ini adalah data nilai indeks harga saham yang diperoleh dalam jangka waktu yang berurutan, data ini dikumpulkan dari waktu ke waktu. Waktu yang digunakan dapat berupa hari bulan, dan tahun.

Analysis time series dilakukan untuk memperoleh pola dari data tersebut dengan menggunakan *historical* data yang akan digunakan untuk meramalkan suatu nilai di masa yang akan datang. Oleh karena itu untuk memilih suatu metode *time series* yang tepat harus mempertimbangkan pola data, agar metode yang digunakan dengan pola tersebut dapat diuji (Bagas, 2019). Dalam melakukan analisis *time series* pertama yang dilakukan mengetahui pola data yang digunakan. Pada umumnya ada 4 tipe pola data pada *time series* yaitu: horizontal, tren, musiman, dan siklis (Yuniar, 2016). Pola horizontal merupakan kejadian yang tidak terduga dan bersifat acak, tetapi kemunculannya dapat mempengaruhi fluktuasi data *time series*. Pola tren merupakan kecenderungan arah data dalam jangka panjang, dapat berupa kenaikan maupun penurunan. Pola musiman merupakan fluktuasi dari data yang terjadi secara periodik dalam kurun waktu satu tahun, seperti triwulan, kuartalan, bulanan, mingguan, atau harian. Sedangkan pola siklis merupakan fluktuasi dari data untuk waktu yang lebih dari satu tahun. Selain itu pemilihan model probabilitas juga penting dilakukan agar sesuai dengan data (Brockwell, 2002).

2.2.4. Recurrent Neural Network

Deep learning mencakup banyak jaringan seperti CNN (*Convolutional Neural Networks*), RNN (*Recurrent Neural Networks*), *Recursive Neural Networks*, DBN (*Deep Belief Networks*) dan banyak lagi (Zhang, 2016).

Recurrent Neural Network (RNN) adalah kasus khusus dari jaringan saraf. Tujuan dari RNN adalah untuk meramalkan langkah berikutnya dalam urutan

pengamatan sehubungan dengan langkah sebelumnya yang diamati dalam urutan tersebut. Pada dasarnya, RNN memanfaatkan observasi sekuensial dan belajar dari tahapan sebelumnya untuk meramalkan / memprediksi tren masa depan. Selama tahap awal, data perlu diingat saat menebak langkah selanjutnya.

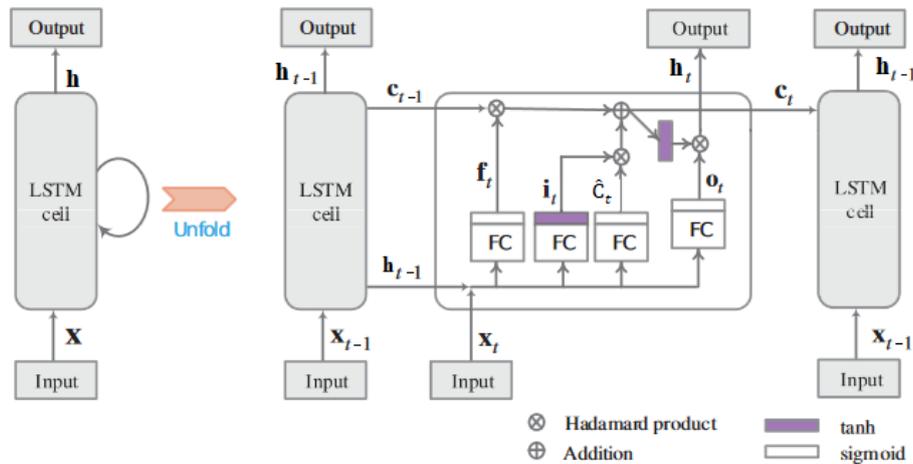
Dalam RNN, lapisan tersembunyi (*hidden layer*) bertindak sebagai penyimpanan internal untuk menyimpan informasi yang dikumpulkan selama tahap awal pemrosesan data sekuensial. Alasan RNN disebut "berulang" adalah karena model melakukan tugas yang sama untuk setiap elemen urutan saat memanfaatkan informasi itu (Ludwig, 2019).

Salah satu tantangan dengan RNN adalah bahwa jaringan ini hanya mengingat beberapa langkah awal dalam urutan data sehingga tidak cocok untuk mengingat urutan yang lebih panjang. Untuk mengatasi kekurangan ini, sehingga jenis jaringan berulang lainnya yaitu LSTM (Moghar, 2020).

2.2.5. Long Short-Term Memory

Di dalam jaringan RNN, informasi berjalan melalui waktu, informasi sebelumnya digunakan sebagai masukan untuk poin berikutnya dapat dihitung di setiap poin. Terdapat dua masalah utama yang terkait dengan jaringan saraf berulang, masalah *vanishing gradient* dan *exploding gradient*. Masalah tersebut dapat diatasi dengan menerapkan *Long Short-Term Memory* (LSTM). LSTM dapat menyimpan informasi sebelumnya dan memperbaharui informasi, dan meneruskan ke lapisan berikutnya tanpa kehilangan informasi. Model LSTM juga memperbaharui kompleksitas setiap bobot dan deret waktu (*time series*) (Brownlee, 2019; Pascanu, 2012; Olah, 2015; Hochreiter, 1997).

Jianjing dkk. (2018) mengutip di "*long short-term memory for machine remaining life prediction*", satu masalah yang terkait dengan RNN standar adalah masalah *fading memory*. Setelah jumlah langkah waktu besar, *timestep* "masa depan" tidak akan berisi memori dari *input* pertama, karena tidak ada struktur dalam lapisan berulang (RNN) standar yang secara individual mengontrol aliran memori itu sendiri (Schmidhuber, 1997). Diagram yang menunjukkan komposisi node-node dari LSTM dapat dilihat pada Gambar 2.1.



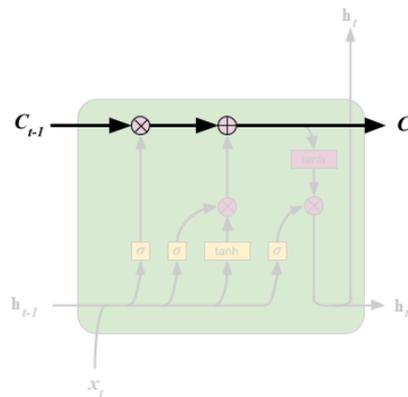
Gambar 2.1. Arsitektur LSTM

Kemampuan menghafal urutan data menjadikan LSTM jenis RNN khusus. Setiap node LSTM sebagian besar terdiri dari satu set sel yang bertanggung jawab untuk menyimpan aliran data yang telah melalui gerbang (*gate*), baris atas di setiap sel menghubungkan model sebagai jalur transportasi data dari masa lalu ke data saat ini, sel independensi membantu model membuang *filter* dari nilai tambahan sel ke sel lainnya. Pada akhirnya, lapisan jaringan saraf *sigmoid* yang menyusun gerbang membawa sel ke nilai yang optimal dengan membuang atau membiarkan data lewat begitu saja. Setiap lapisan *sigmoid* memiliki nilai 0 atau 1. Tujuannya adalah untuk mengontrol keadaan setiap sel, gerbang dikontrol sebagai berikut:

- *Forget Gate* menghasilkan angka 0 dan 1, di mana 1 berarti ingat; sedangkan, 0 menunjukkan bahwa data harus dilupakan;
- *Memory Gate* memilih data baru mana yang akan disimpan di dalam sel. Pertama, lapisan *sigmoid input layer* memilih nilai mana yang akan diubah. Selanjutnya, lapisan *tanh* membuat vektor nilai kandidat baru yang dapat ditambahkan ke *state*;
- *Output Gate* memutuskan apa yang akan menjadi *output* dari setiap sel. Nilai keluaran akan didasarkan pada status sel bersama dengan data yang telah disaring dan data terbaru (Moghar, 2020).

Ide kunci LSTM adalah jalur yang menghubungkan konteks lama (h_{t-1}) ke konteks baru (h_t) yang terletak di garis horizontal atas modul LSTM, seperti pada

Gambar 2.2. Jalur tersebut sangat berguna karenanya suatu nilai pada konteks lama dapat mudah diteruskan ke konteks baru dengan modifikasi yang minim, jika diperlukan.



Gambar 2.2. Mekanisme Konteks Memori LSTM

Konteks merupakan suatu vektor yang jumlah elemennya ditentukan sendiri. Masing-masing elemen diharapkan dapat merekam suatu fitur *input* yang fitur-fitur ini akan ditemukan sendiri oleh LSTM dalam proses latihan. Dalam LSTM, memori tahapan sebelumnya dapat dilakukan di sepanjang gerbang garis memori yang terlihat pada Gambar 2.2.

Gerbang *sigmoid* merupakan ide kunci lain yang bertugas mengatur seberapa banyak informasi yang dapat dilewati pada gerbang tersebut. *Output* dari gerbang *sigmoid* akan dikalikan dengan suatu nilai lain untuk mengontrol seberapa banyak nilai/informasi dari c_{t-1} tersebut digunakan menjadi c_t . Fungsi aktivasi *sigmoid* dapat dijabarkan dalam persamaan (2.1).

$$\sigma(x) = \frac{1}{(1 + e^{-x})} \quad (2.1)$$

Dimana x = data input, dan e = konstanta matematika (2,718281828...).

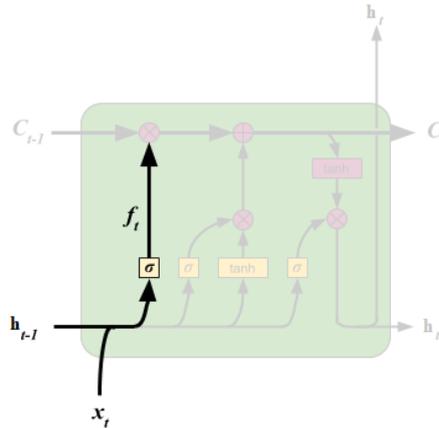
Dan fungsi aktivasi *tanh* dapat dijabarkan dalam persamaan (2.2).

$$\tanh(x) = \sigma(2x) - 1 \quad (2.2)$$

2.2.5.1. Forget Gate

Pada persamaan (2.3), h_{t-1} dan x_t akan melewati gerbang *sigmoid* pada Gambar 2.3. Langkah pertama LSTM memutuskan informasi apa yang akan

dihapus/dibuang dari *cell state*. Keputusan ini dibuat oleh *sigmoid layer* yang disebut *forget gate layer* akan memproses h_{t-1} dan x_t sebagai *input*, dan menghasilkan *output* berupa angka 0 dan 1 pada *cell state* (c_t).



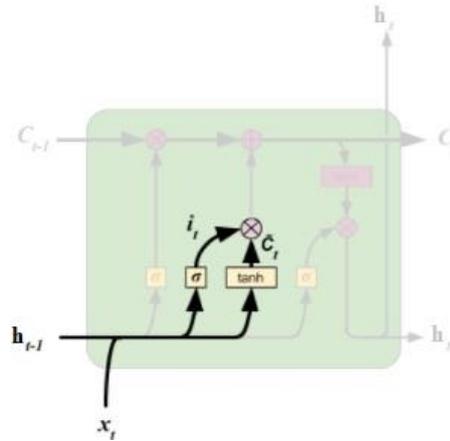
Gambar 2.3. *Forget Gate*

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f) \tag{2.3}$$

Dimana $f_t = \textit{forget gate}$, $\sigma = \textit{fungsi sigmoid}$, $W_f = \textit{nilai weight untuk forget gate}$, $h_{t-1} = \textit{nilai output sebelum orde ke-t atau konteks lama}$, $x_t = \textit{nilai input pada orde ke-t}$, $[h_{t-1}, x_t] = \textit{merupakan operasi konkatenasi, yaitu penggabungan dari } h_{t-1} \textit{ dan } x_t$, $b_f = \textit{nilai bias pada forget gate}$.

2.2.5.2. *Input Gate*

Langkah selanjutnya dapat dilihat pada Gambar 2.4, yaitu memutuskan informasi baru apa yang akan disimpan di *cell state* (c_t) pada persamaan (2.4). Ada dua bagian, yaitu lapisan *sigmoid* yang disebut *input gate layer* (i_t) menentukan nilai mana yang akan diperbaharui.



Gambar 2.4. *Input Gate*

$$\hat{I}_t = \sigma(W_i [h_{t-1}, x_t] + b_i) \quad (2.4)$$

Dimana $\hat{I}_t = \text{input gate}$, $\sigma = \text{fungsi sigmoid}$, $W_i = \text{nilai weight untuk input gate}$, $h_{t-1} = \text{nilai out put sebelum orde ke-}t$, $b_i = \text{nilai bias pada input gate}$.

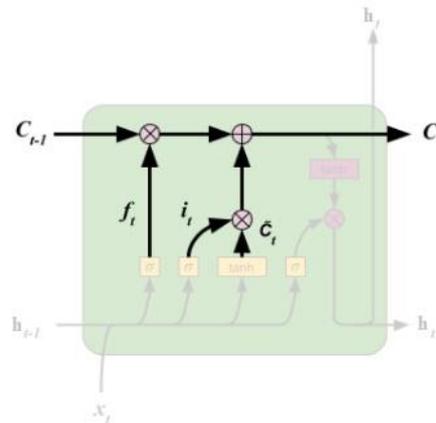
Selanjutnya, lapisan *tanh* membuat vektor nilai baru, \hat{C}_t yang dapat ditambahkan ke *cell state* (c_t) pada persamaan (2.5). Pada langkah selanjutnya, keduanya digabungkan untuk pembaharuan *cell state*.

$$\hat{C}_t = \tanh(W_c [h_{t-1}, x_t] + b_c) \quad (2.5)$$

Dimana $\hat{C}_t = \text{nilai baru yang dapat ditambahkan ke cell state}$, $\tanh = \text{fungsi tanh}$, $W_c = \text{nilai weight untuk cell state}$, $h_{t-1} = \text{nilai output sebelum orde ke-}t$, $x_t = \text{nilai input pada orde ke-}t$, $b_c = \text{nilai bias untuk cell state}$.

2.2.5.3. *Memory Update Gate*

Selanjutnya untuk memperbaharui *cell state* lama pada persamaan (2.6), c_{t-1} , ke *cell state* baru c_t . Dengan mengalikan keadaan *state* lama dengan f_t , diputuskan untuk dilupakan pada *forget gate layer*. Lalu ditambahkan $\hat{I}_t \otimes \hat{C}_t$. Ini adalah nilai baru, diskalakan dengan berapa banyak untuk diputuskan pembaharuan setiap *cell state* dapat dilihat pada Gambar 2.5.



Gambar 2.5. Memory Update Gate

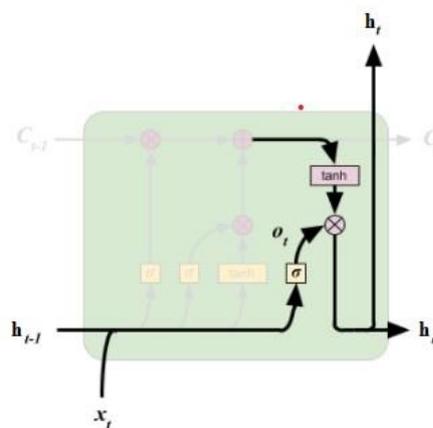
$$c_t = f_t \otimes c_{t-1} \oplus i_t \otimes \hat{c}_t \quad (2.6)$$

Dimana $c_t = cell\ state$, $f_t = forget\ gate$, $c_{t-1} = cell\ state\ sebelum\ orde\ ke-t$, $i_t = input\ gate$, $\hat{c}_t = nilai\ baru\ yang\ dapat\ ditambahkan\ ke\ cell\ state$.

2.2.5.4. Output Gate

Langkah terakhir dari LSTM terjadi pada Gambar 2.6 menggunakan persamaan (2.7), dengan menjalankan lapisan *sigmoid* yang menentukan bagian *cell state* mana yang menjadi *output*.

$$O_t = \sigma (W_o [h_{t-1}, x_t] + b_o) \quad (2.7)$$



Gambar 2.6. Output Gate

Kemudian, menempatkan *cell state* melalui fungsi aktivasi *tanh* (untuk mengubah nilai antara -1 dan 1) dan mengalikannya dengan *output* dari gerbang *sigmoid*, sehingga hanya mengeluarkan bagian-bagian yang diputuskan, dan terjadi pada persamaan (2.8).

$$h_t = O_t \otimes \tanh(C_t) \quad (2.8)$$

Dimana $O_t = \text{output gate}$, $\sigma = \text{fungsi sigmoid}$, $W_o = \text{nilai weight untuk output gate}$, $h_{t-1} = \text{nilai output sebelum orde ke-}t$, $x_t = \text{nilai input pada orde ke-}t$, $b_o = \text{nilai bias pada output gate}$, $h_t = \text{nilai output orde ke-}t$, $\tanh = \text{fungsi tanh}$, $c_t = \text{cell state}$.

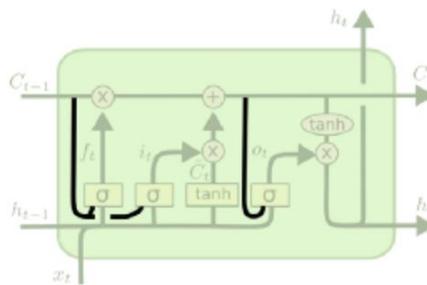
2.2.6. Peephole Connection LSTM

Seiring dengan waktu LSTM mengalami perkembangan, salah satunya adalah *Peephole Connection* atau lubang intip yang dipopulerkan pada tahun 2002 oleh Schmidhuber dan Felix dkk., pada tahun 2002 mengutip dalam “*Learning Precise Timing with LSTM Recurrent Networks, Peephole Connections*” memungkinkan semua gerbang untuk memeriksa status sel saat ini bahkan ketika *output gate* ditutup. Informasi ini menjadi penting untuk menemukan solusi *network* yang berfungsi dengan baik.

Dalam LSTM konvensional atau biasa disebut *Vanilla LSTM*, setiap *gate* adalah *input LSTM* terdiri dari dua bagian, yaitu *input* x_t waktu saat ini (t) dan merupakan *output* pada waktu $t - 1$, jika *output* dimatikan pada waktu t , *output* jaringan pada waktu t akan menjadi 0. Sehingga *input* dari gerbang akan terhubung dengan *input* x_t dan informasi historisnya akan hilang. Kekurangan ini dapat menyebabkan hilangnya sejumlah besar informasi penting, yang akan mempengaruhi kinerja jaringan berulang *Vanilla LSTM*.

Lei (2020) mengutip dalam “*Time Series-oriented Load Prediction Using Deep Peephole LSTM*”, solusi sederhana dan efektif untuk pembatasan tersebut adalah menambahkan koneksi lubang intip ke semua gerbang di jaringan LSTM yang sama (Felix, 2002). Sehingga *Peephole Connection* menjadi lubang intip oleh *cell state* (c_t) untuk melihat *cell state* yang terdapat pada sel LSTM sebelumnya agar tidak

hanya bergantung pada *hidden state* sebelumnya (h_{t-1}). Ting Yang, dkk. 2018 mengutip dalam jurnal berjudul “*Novel Method of Wind Speed Prediction by Peephole Connection LSTM*”, bahwa lubang intip ini sebagai monitor langsung yang memungkinkan semua pintu untuk memeriksa status sel saat ini (Bottieu, 2018). *Hidden state* (h_t) pada *Vanilla LSTM* dan *Peephole Connection LSTM* tetap sama, hanya saja *cell state* saat ini (c_t) tidak hanya dapat melihat *hidden state* sebelumnya (h_{t-1}) namun juga dapat melihat c_{t-1} .



Gambar 2.7. *Peephole Connection LSTM*

Peephole Connection dari *cell state* (c_t) menuju ke tiga gerbang yaitu i , f , o pada Gambar 2.7. Koneksi lubang intip ini merupakan kontribusi aktivasi sel memori pada langkah waktu ($t - 1$), yaitu kontribusi dari c_{t-1} . Seperti yang ditunjukkan pada Gambar 2.7, semua gerbang dapat mendeteksi keadaan sel saat ini bahkan ketika *output gate* ditutup. Oleh karena itu *forget gate*, *input gate*, dan *output gate*, semuanya akan dipengaruhi oleh status sel sebelumnya (c_{t-1}), dan diperbaharui menggunakan persamaan (2.9), (2.10), dan (2.11) (Fu, 2020).

$$f_t = \sigma (W_f [c_{t-1}, h_{t-1}, x_t] + b_f) \quad (2.9)$$

$$i_t = \sigma (W_i [c_{t-1}, h_{t-1}, x_t] + b_i) \quad (2.10)$$

$$O_t = \sigma (W_o [c_{t-1}, h_{t-1}, x_t] + b_o) \quad (2.11)$$

Dimana $f_t = \text{forget gate}$, $i_t = \text{input gate}$, $O_t = \text{output gate}$, $\sigma = \text{fungsi sigmoid}$, $W_f = \text{nilai weight untuk forget gate}$, $W_i = \text{nilai weight untuk input gate}$, $W_o = \text{nilai weight untuk output gate}$, $c_{t-1} = \text{cell state sebelum orde ke-}t$, $h_{t-1} = \text{nilai output}$

sebelum *orde* ke- t , x_t = nilai *input* pada *orde* ke- t , b_f = nilai bias pada *forget gate*, b_i = nilai bias pada *input gate*, b_o = nilai bias pada *output gate*.

2.2.7. Facebook's Prophet

Facebook membuat *Prophet* menjadi *library open source* yang dibuat oleh Facebook's Core Data Science team, merupakan metode *forecasting* yang tersedia dalam bahasa pemrograman *Python* dan *R*. Peramalan/prediksi merupakan ilmu data yang penting bagi banyak aktivitas dalam suatu organisasi. Menghasilkan prakiraan berkualitas tinggi bukanlah masalah yang mudah bagi mesin atau sebagian besar analis.

Prophet sepenuhnya otomatis yang digabungkan dengan *analyst-in-the-loop* sehingga tidak akan menghasilkan peramalan yang tidak memuaskan. Dalam data *timeseries*, analis dapat meningkatkan ketepatan peramalan dengan mengubah berbagai parameter yang mudah diinterpretasikan. Gagasan penting dalam *Prophet* adalah bahwa dengan melakukan pekerjaan yang lebih baik dalam menyesuaikan komponen tren dengan sangat fleksibel, *Prophet* dapat memodelkan musim dengan lebih akurat dan hasilnya adalah peramalan yang lebih akurat.

Secara default, *Prophet* akan memberikan interval ketidakpastian untuk komponen tren dengan mensimulasikan perubahan tren di masa mendatang pada data *timeseries*. Untuk membuat model ketidakpastian tentang efek *seasonality* atau liburan di masa mendatang, dapat menggunakan beberapa ratus iterasi HMC (yang membutuhkan waktu beberapa menit) dan peramalan akan menyertakan peramalan ketidakpastian *seasonality* (Taylor, 2017).

Wen-Xiang dkk., pada tahun 2019 mengutip dalam jurnal "*Combine Facebook Prophet and LSTM with BPNN Forecasting financial markets : the Morgan Taiwan Index*" bahwa *Prophet* memiliki kemampuan pemrosesan yang sangat baik untuk meramalkan data yang memiliki musiman sangat banyak (Upadhayula, 2019). Facebook telah mengadopsi *Prophet* sebagai aplikasinya, dan telah menerapkan perubahan pada parameternya di perangkat lunak *R* dan pengguna terbuka *Python*.

Berdasarkan data *timeseries* yang diuraikan oleh Harvey dan Peter pada tahun 1990 dalam jurnal "*Estimation procedures for structural time series models*"

dengan 3 model komponen, yaitu *trend*, *seasonality*, dan *holidays* yang digabungkan dalam persamaan berikut.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (2.12)$$

Dimana $g(t)$ = fungsi tren yang memodelkan perubahan non-periodik dalam nilai *timeseries*, $s(t)$ = mewakili perubahan periodik (misalnya, musiman mingguan dan tahunan), $h(t)$ = mewakili efek liburan yang terjadi pada jadwal yang berpotensi tidak teratur satu hari atau lebih, ϵ_t = mewakili setiap perubahan istimewa yang tidak diakomodasi oleh model.

Sean (2017) mengutip dalam *preprints* dengan judul *forecasting at scale* bahwa, spesifikasi ini mirip dengan *generalized additive model* (GAM) (Hastie, 1987), kelas model regresi dengan kemungkinan penghalusan non-linier diterapkan ke regulator. *Prophet* menggunakan waktu sebagai regressor tetapi mungkin beberapa fungsi waktu linier dan non-linier sebagai komponen. Pemodelan musiman sebagai komponen aditif adalah pendekatan yang sama yang dilakukan oleh pemulusan eksponensial (Gardner, 1985).

Meskipun penggunaan model generatif seperti ARIMA memberikan beberapa keuntungan inferensial, namun *Prophet* memberikan beberapa kelebihan yang lebih praktis, seperti:

1. Fleksibilitas mudah mengakomodasi musim dengan beberapa periode dan membiarkan analis membuat asumsi yang berbeda mengenai tren;
2. Berbeda dengan model ARIMA, pengukuran tidak perlu diberi jarak secara teratur, dan tidak perlu menginterpolasi *missing value* misalnya dari menghilangkan pencilan.
3. *Fitting* sangat cepat, memungkinkan analis untuk secara interaktif mengeksplorasi banyak spesifikasi model, misalnya dalam aplikasi Shiny.
4. Model peramalan memiliki parameter yang mudah diinterpretasikan yang dapat diubah oleh analis untuk memaksakan asumsi pada ramalan. Selain itu, analis biasanya memiliki pengalaman dengan regresi dan dengan mudah dapat memperluas model untuk memasukkan komponen baru.

2.2.7.1. Trend Model

Terdapat dua model tren yang dapat diimplementasikan pada aplikasi Facebook, yaitu *saturating growth model*, dan *a piecewise linear model*.

1. Nonlinear, Saturating Growth Model

Untuk peramalan pertumbuhan, komponen inti dari proses menghasilkan data adalah model bagaimana populasi tumbuh dan bagaimana populasi diharapkan untuk terus berkembang. Pertumbuhan semacam ini biasanya dimodelkan dengan menggunakan model pertumbuhan logistik, yang dihitung menggunakan persamaan (2.13).

$$g(t) = \frac{C}{1+e^{(-k(t-m))}} \quad (2.13)$$

Dimana C = populasi kasus, k = laju pertumbuhan, dan m = parameter *offset*.

Terdapat 2 aspek penting yang tidak tercakup dalam persamaan (2.13), yaitu kapasitas (C) dimana jumlah kapasitas. Dengan demikian C berubah bergantung dengan waktu (t) menjadi $C(t)$. Selanjutnya, laju pertumbuhan k juga tidak konstan sehingga k bergeser sejauh *change point* (S) menjadi *change point* berdasarkan waktu (s_j) disebut δ yang bergerak dari $j = 1, \dots, S$. Dengan vektor laju pertumbuhan $\delta \in \mathbb{R}^S$, dimana δ_j adalah perubahan dalam laju pertumbuhan berdasarkan waktu s_j . Sehingga *base rate* (k) dapat dirumuskan dalam $k + \sum_{j:t>s_j} \delta_j$, singkatnya δ bergerak dari $j:t > s_j$ yang dapat direpresentasikan sebagai vektor $\mathbf{a}(t) \in \{0, 1\}^S$, dimana

$$\mathbf{a}_j(t) = \begin{cases} 1, & \text{jika } t \geq s_j, \\ 0, & \text{lainnya.} \end{cases}$$

Dan laju pertumbuhan pada waktu ke (t) menjadi $k + \mathbf{a}(t)^T \delta$. Dan ketika k mengalami pergeseran, maka *offset parameter* (m) juga harus digeser untuk menghubungkan akhir segmen. Perubahan nilai berdasarkan *change point* j didefinisikan sebagai

$$\gamma_j = \left(s_j - m - \sum_{l < j} \gamma_l \right) \left(1 - \frac{k + \sum_{l < j} \delta_l}{k + \sum_{l \leq j} \delta_l} \right) \quad (2.14)$$

Sehingga persamaan model *piecewise growth* dapat dilihat pada persamaan (2.15).

$$g(t) = \frac{c(t)}{1 + e^{-(k + \mathbf{a}(t)^T \boldsymbol{\delta})(t - (m + \mathbf{a}(t)^T \boldsymbol{\gamma}))}} \quad (2.15)$$

2. Linear Trend with Changepoints

Untuk masalah peramalan yang tidak menunjukkan *saturating growth*, laju pertumbuhan yang konstan, sebagian memberikan model yang *parsimonious* dan sering kali berguna. Berikut persamaan yang dapat digunakan.

$$g(t) = (k + \mathbf{a}(t)^T \boldsymbol{\delta})t + (m + \mathbf{a}(t)^T \boldsymbol{\gamma}) \quad (2.16)$$

Dimana k = adalah laju pertumbuhan, $\boldsymbol{\delta}$ = penyesuaian laju, m = parameter *offset*, dan γ_j disetel ke $-s_j \delta_j$ untuk menjadikan model dalam bentuk kontinu.

2.2.7.2. Seasonality

Deret waktu bisnis sering kali memiliki musim *multi-period* banyak periode musiman sebagai akibat dari perilaku manusia yang diwakili. Misalnya, 5 hari kerja dalam seminggu dapat menghasilkan efek pada rangkaian waktu yang berulang setiap minggu, sementara jadwal liburan dapat menghasilkan efek yang berulang setiap tahun. Untuk menyesuaikan dan meramalkan efek ini, peneliti harus menentukan model musiman yang sering menjadi fungsi periodik dari T .

Prophet mengandalkan deret Fourier untuk memberikan model efek periodik yang fleksibel (Harvey, 1993). Misalkan P menjadi periode reguler yang diharapkan dari deret waktu (misalnya $P = 362,25$ untuk data tahunan atau $P = 7$ untuk data mingguan, ketika menskalakan variabel waktu dalam sehari). Perkiraan efek *smooth seasonal* dengan deret *Fourier* standar dapat dirumuskan dengan persamaan (2.17).

$$s(t) = \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \quad (2.17)$$

Fitting musiman membutuhkan estimasi parameter $2N \beta = [a_1 b_1 \dots a_N b_N]^T$. Ini dilakukan dengan membuat matriks vektor musiman untuk setiap nilai t dalam data historis dan masa depan, misalnya dengan musiman tahunan dan $N = 10$,

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{365.25}\right), \dots, \sin\left(\frac{2\pi(10)t}{365.25}\right) \right] \quad (2.18)$$

Dengan komposisi *seasonal* dirumuskan dalam persamaan (2.19).

$$s(t) = X(t) \beta \quad (2.19)$$

Dengan β merupakan bobot dari $X(t)$.

Untuk *season* tahunan dan mingguan, menggunakan $N = 10$ untuk *season* tahunan dan $N = 3$ untuk *season* mingguan sebagai nilai *default* dan masing-masing dapat bekerja dengan baik untuk sebagian besar masalah.

2.2.7.3. Holidays dan Events

Liburan dan suatu peristiwa memberikan pengaruh besar dan dapat diramalkan untuk banyak *time series* dalam dunia bisnis dan biasanya tidak mengikuti pola periodik, sehingga efeknya tidak dimodelkan dengan baik oleh model. Misalnya, peringatan *Thanksgiving* di Amerika Serikat terjadi pada hari Kamis keempat di bulan November. *Super Bowl*, salah satu acara televisi terbesar di Amerika Serikat, terjadi pada hari Minggu di bulan Januari atau Februari yang sulit diumumkan secara terprogram. Banyak negara di seluruh dunia memiliki hari libur besar yang mengikuti kalender lunar. Dampak hari libur tertentu pada *time series* biasanya serupa tahun demi tahun, jadi penting untuk memasukkannya ke dalam peramalan masa depan.

Tabel 2.1 Contoh daftar hari libur di Amerika Serikat

<i>Holiday</i>	Negara	Tahun	Tanggal
Thanksgiving	US	2015	26-Nov-15
Thanksgiving	US	2016	24-Nov-16
Thanksgiving	US	2017	23-Nov-17
Thanksgiving	US	2018	22-Nov-18
Christmas	*	2015	25-Dec-15
Christmas	*	2016	25-Dec-16
Christmas	*	2017	25-Dec-17
Christmas	*	2018	25-Dec-18

Sumber: PeerJ Preprints 2017

Tabel 2.1 menunjukkan contoh daftar hari libur di Amerika Serikat. Negaranya dispesifikkan karena hari libur di setiap negara berbeda-beda. Dalam kasus peramalan, *Prophet* menggunakan gabungan rangkaian hari libur global dan negara-negara tertentu.

Daftar hari libur yang digabungkan dan dimasukkan ke dalam model yang dibuat. Untuk setiap *holiday* i , misalkan D_i adalah tanggal di tahun sebelumnya dan yang akan datang untuk hari libur tersebut, dimana i berjalan dari 1 hingga $L = \text{jumlah } holiday D_i$. *Prophet* menambahkan fungsi indikator yang mewakili apakah waktu (t) merupakan tanggal dari *holiday* i , dan menetapkan di setiap *holiday* parameter k_i yang sesuai dengan perubahan dalam peramalan dan dirumuskan pada persamaan (2.21). Ini dilakukan dengan cara yang mirip dengan *seasonality* dengan membuat matrix regresi seperti pada persamaan (2.20).

$$Z(t) = [\mathbf{1}(t \in D_1), \dots, \mathbf{1}(t \in D_L)] \quad (2.20)$$

Dengan

$$h(t) = Z(t)\mathbf{k} \quad (2.21)$$

sama dengan *seasonality*, digunakan prior $\mathbf{k} \sim \text{Normal}(0, v^2)$.

Contoh ($t \in D_L$) :

$$t(D_1(\text{holiday } \textit{thanksgiving})) = \begin{bmatrix} 26 \text{ Nov } 2015 \\ 24 \text{ Nov } 2016 \\ 23 \text{ Nov } 2017 \\ 22 \text{ Nov } 2018 \end{bmatrix}$$

$$t (D_2(\text{holiday christmas})) = \begin{bmatrix} 25 \text{ Dec } 2015 \\ 25 \text{ Dec } 2016 \\ 25 \text{ Dec } 2017 \\ 25 \text{ Dec } 2018 \end{bmatrix}$$

Contoh $h(t)$:

$$h(t) = \left(\mathbf{1} \begin{bmatrix} 26 \text{ Nov } 2015 \\ 24 \text{ Nov } 2016 \\ 23 \text{ Nov } 2017 \\ 22 \text{ Nov } 2018 \end{bmatrix}, \mathbf{1} \begin{bmatrix} 25 \text{ Dec } 2015 \\ 25 \text{ Dec } 2016 \\ 25 \text{ Dec } 2017 \\ 25 \text{ Dec } 2018 \end{bmatrix} \right) \times \mathbf{k}$$

Seringkali penting untuk menyertakan hari-hari di sekitar hari libur, seperti pada akhir pekan *Thanksgiving*. Untuk memperhitungkannya, dapat disertakan parameter tambahan untuk hari-hari sebelum atau sesudah *holiday*, sebagai *holiday* itu sendiri.

2.2.8. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) merupakan salah satu teknik pengukur akurasi terhadap variabel kontinu dengan menjumlahkan selisih dua larik, yaitu y dan \hat{y} . MAE digunakan untuk mengukur rata-rata *error* untuk meminimumkan *error*. MAE dirumuskan sebagai berikut:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \tag{2.22}$$

Dimana n adalah jumlah data, y_t adalah nilai pelatihan ke $-t$, dan \hat{y}_t merupakan nilai pengujian ke $-t$ (Rizal, 2020).

2.2.9. Root Mean Square Error (RMSE)

Salah satu teknik yang digunakan untuk mengevaluasi hasil peramalan yaitu *Root Mean Square Error* (RMSE). RMSE merupakan pengecekan kesalahan yang membandingkan nilai sesungguhnya dengan nilai yang didapatkan dari pengujian dengan hasil dinyatakan dengan nilai mutlak. RMSE tersebut dirumuskan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \tag{2.23}$$

Dimana n adalah jumlah data, y_t adalah data pelatihan, dan \hat{y}_t adalah data pengujian ke $-t$ (Bagas, 2019).

2.2.10. Normalisasi Data

Normalisasi data adalah proses membuat beberapa variabel memiliki rentang nilai yang sama, tidak ada yang terlalu besar maupun terlalu kecil sehingga dapat membuat analisis statistik menjadi lebih mudah dan juga mempercepat proses *train*. Metode normalisasi data yang digunakan dalam penelitian ini adalah *Min-Max Scaling Normalization*. Persamaan umum untuk *scaling* dalam rentang $[0, 1]$ adalah sebagai berikut.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.24)$$

Dimana x = nilai asli dan x' = nilai yang dinormalisasi.