# 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 rancagan 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 | *Program Tool* | IDLE Python |
| 3 | *Framework Python* | Dash, Pandas, Yfinance, Numpy. |
| 4 | *Text Editor* | Visual studio code |

### 4.2 *Source Code* Objek LSTM

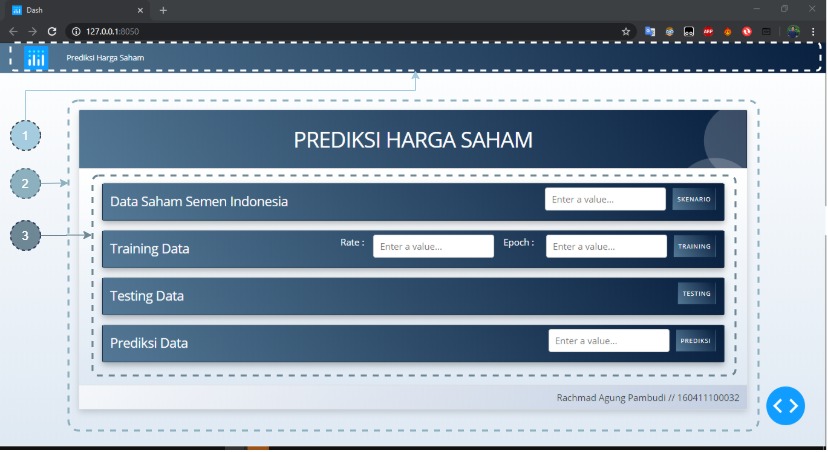
Pada tahap ini akan diimplementasikan *Source code* dari proses forward, backward, training, dan testing.

|  |
| --- |
| 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.

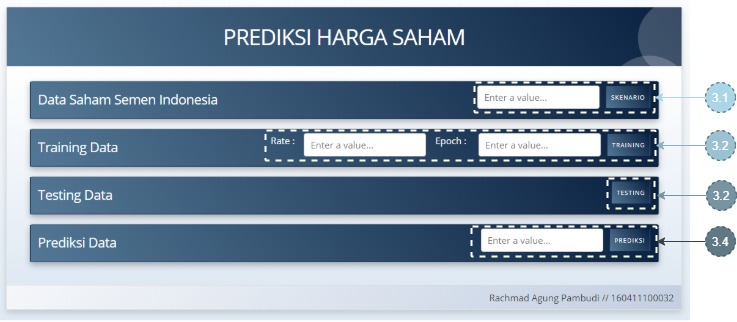
### 4.3.1 Tampilan Aplikasi



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* dantomboluntuk 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 1tombol 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 1tombol 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 1tombol 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

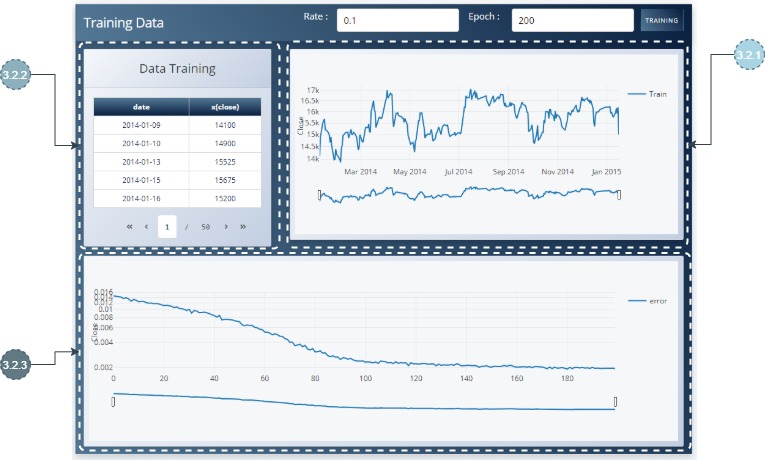
Dari *Gambar 4.3*. Tampilan tabel dan grafik full data close saham dibagi menjadi 2 yaitu:

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 1tombol 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 menjalan 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 menunjukan tingkat errornya.

#### 4.3.3.3 Tampilan Testing data

Gambar 4.25 Tampilan Testing Data

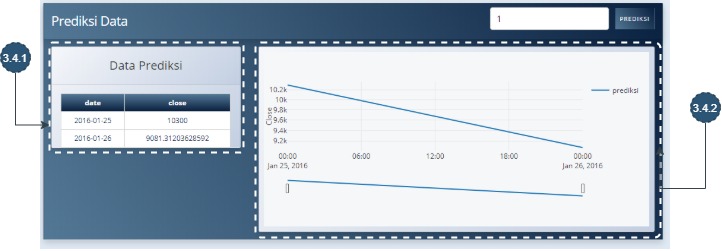
Dari *Gambar 4.5*. Tampilan Testing data dibagi menjadi 4 konten sesuai dengan fungsinya yaitu:

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 menunjukan 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 prediskidalam 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 melalu 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 paremeter yang terdapat pada tebel 4.5 mengcapai 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 melalu 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 |

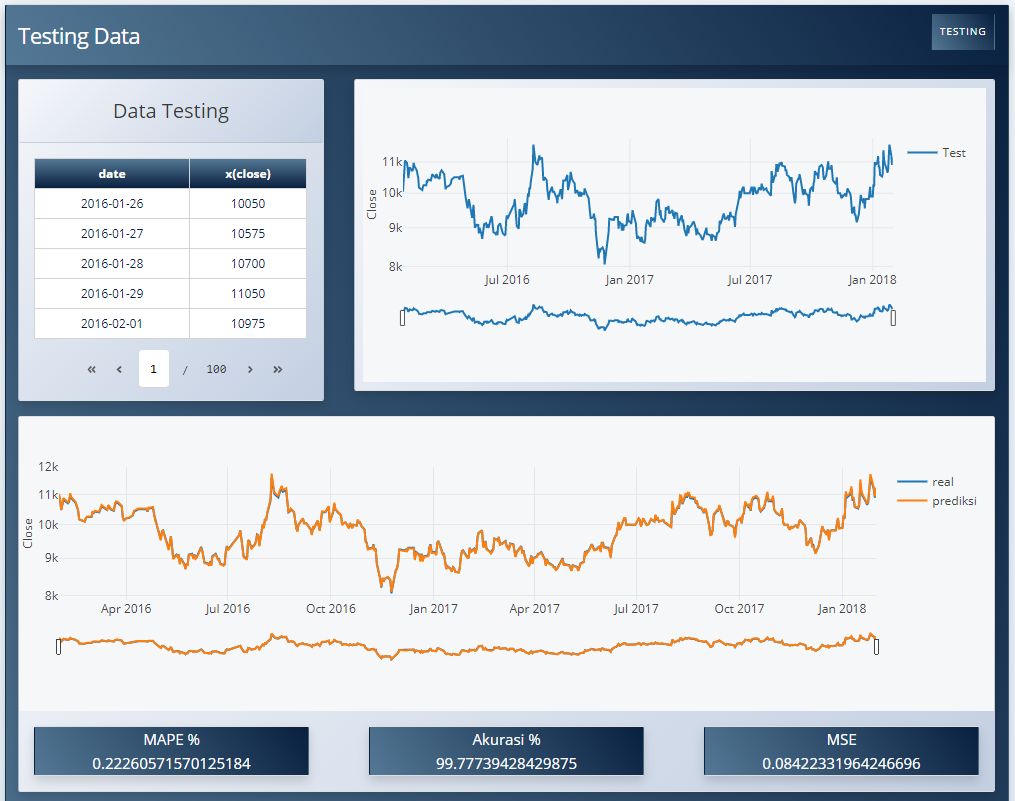
Table 49 PerubahanUji Coba dengan perubahan nilai Learning rate pada data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |

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

Table 50 Hasil terbaik Skenario 1

Akurasi error yang dihasilkan dengan menggunakan paremeter yang terdapat pada tebel 4.10 mengcapai 99.68%.

Gambar 4.29. Tampilan hasil testing data scenario 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 melalu 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 paremeter yang terdapat pada tebel 4.14 mengcapai 99.68%.



Gambar 4.10 Tampilan hasil testing data scenario 3

# BAB V PENUTUP

## 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.

## 5.2 Saran

Saran yang diajukan untuk penelitan 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 Neural Network algoritma LSTM.

# 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. Mubarok, “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#/