



UNIVERSITAS
GADJAH MADA

Aplikasi Sains Komputasional: Prediksi Saham Menggunakan Metode *Long- Short Term Memory (LSTM)*

Ivan Luthfi Ihwani, M. Sc.



UNIVERSITAS
GADJAH MADA

Pasar Modal

- Di dalam undang-undang, pasar modal didefinisikan sebagai kegiatan yang berhubungan dengan penawaran umum dan perdagangan efek, perusahaan publik yang berkaitan dengan efek yang diterbitkannya, serta lembaga dan profesi yang berkaitan dengan efek. (Bab 1, Pasal 1, angka 13, UURI no. 8, 1995, tentang pasar modal)
- Pasar Modal:
 - Sebagai sarana bagi perusahaan untuk mendapatkan dana (pendanaan usaha)
 - Sebagai sarana investasi bagi masyarakat (pemodal)



UNIVERSITAS
GADJAH MADA

Investasi dan Saham

- Investasi adalah komitmen atas sejumlah dana atau sumber daya lainnya yang dilakukan pada saat ini, dengan tujuan memperoleh keuntungan di masa yang akan datang.
- Saham adalah surat berharga yang berbentuk sertifikat guna menunjukkan bukti kepemilikan suatu perusahaan.



UNIVERSITAS
GADJAH MADA

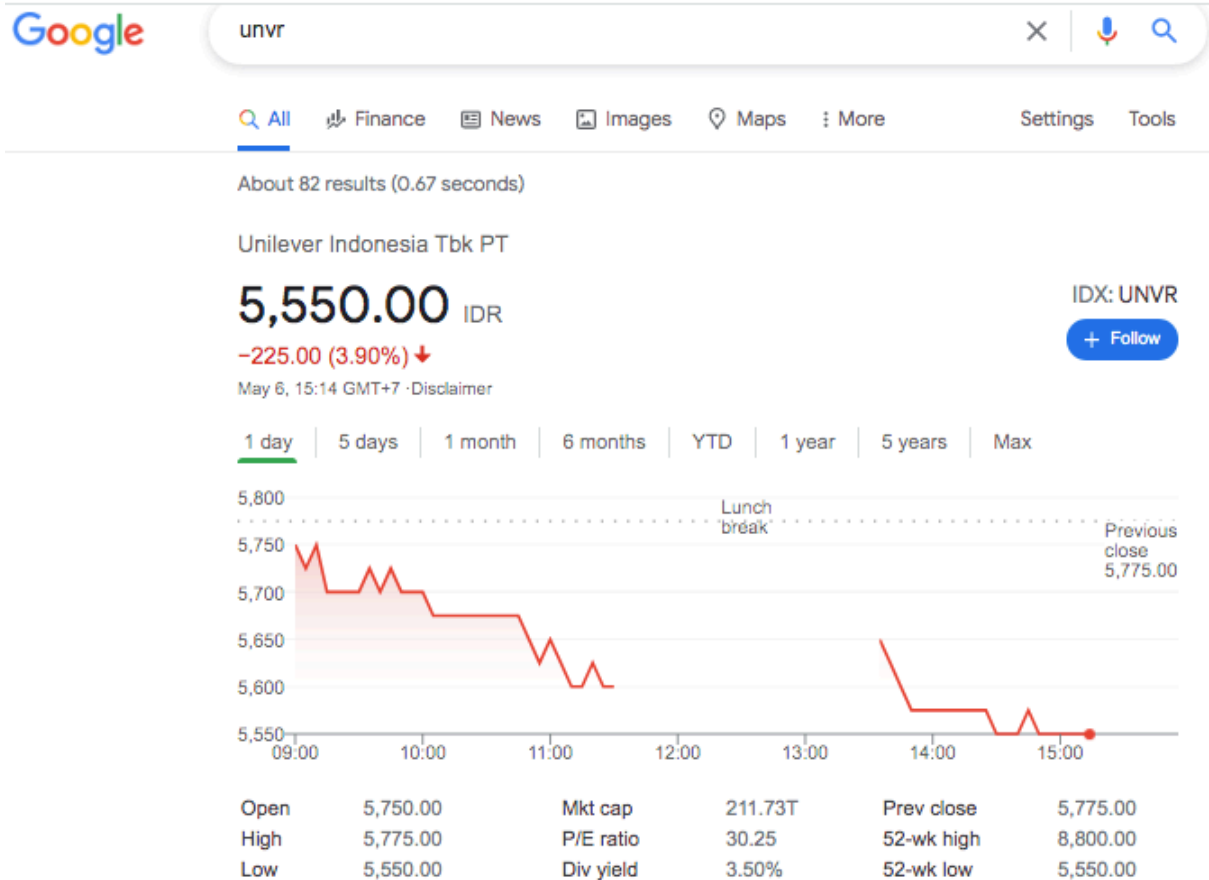
Harga Saham

- Harga saham adalah nilai bukti penyertaan modal pada perseroan terbatas yang telah terdaftar di bursa efek Indonesia (BEI).
- Harga saham dibentuk dari interaksi antara para penjual dan pembeli saham yang dilatarbelakangi oleh harapan mereka terhadap keuntungan perusahaan.
- Terdapat 4 komponen harga saham harian, yaitu pembukaan (*open*), harga tertinggi (*high*), harga terendah (*low*), harga penutupan (*close*), dan harga penutupan yang sudah termasuk harga setelah *stock split* dan pembagian deviden (*adjusted close*).



UNIVERSITAS
GADJAH MADA

Contoh



Sumber: <https://google.com>



UNIVERSITAS
GADJAH MADA

Analisa Penilaian Harga Saham

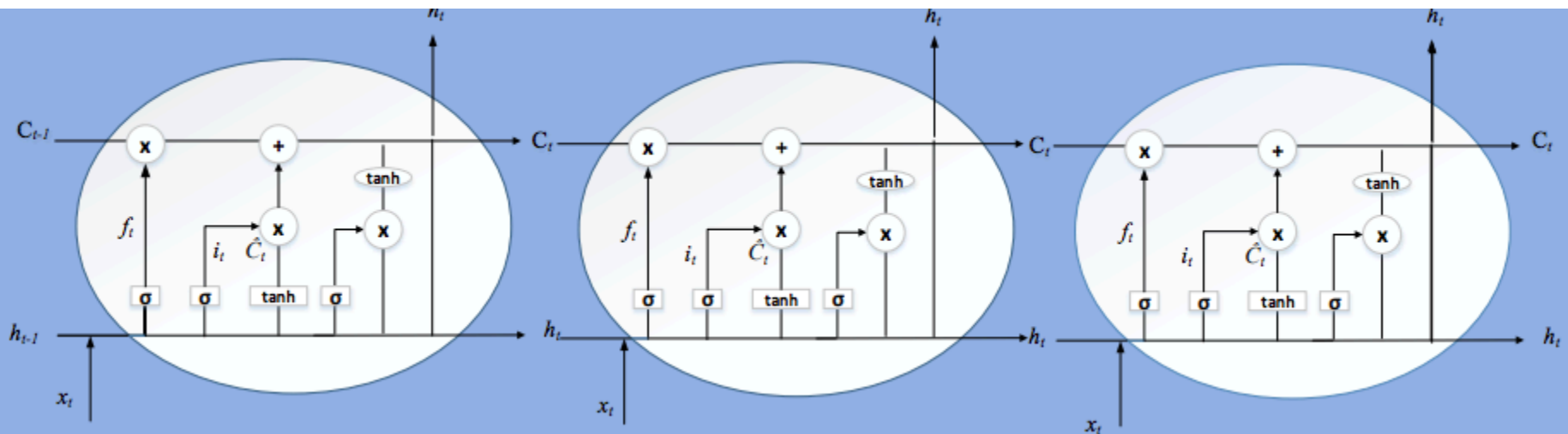
- Ada 2 macam pendekatan yang digunakan untuk menganalisa harga saham, yaitu:
 - 1) **Analisis Fundamental:** menggunakan data fundamental yang berasal dari keuangan perusahaan. Misal: pendekatan deviden, pendekatan Price Earning Ratio (PER)
 - 2) **Analisis Teknikal:** menggunakan data pasar dari saham untuk menentukan nilai saham.



UNIVERSITAS
GADJAH MADA

Metode LSTM

Long Short Term Memory (LSTM) merupakan salah satu pengembangan *neural network* yang dapat digunakan untuk pemodelan data *time series*.



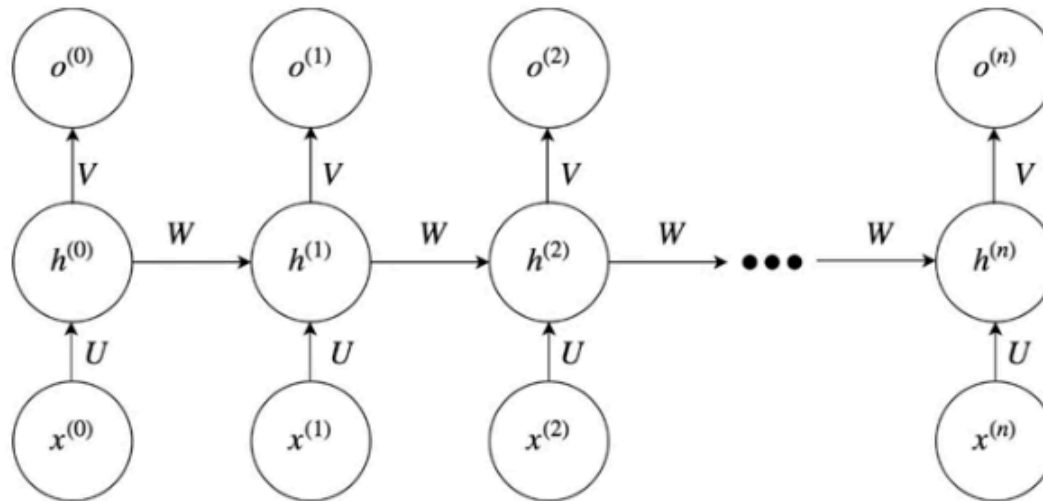
Sumber:

- S. Hochreiter and J. Schmidhuber, 1997.
- Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins, 1999



RNN

Jaringan saraf tiruan berulang atau recurrent neural network (RNN) adalah jenis arsitektur jaringan saraf tiruan yang pemrosesannya dipanggil berulang-ulang untuk memproses masukan yang biasanya adalah data sekuensial.

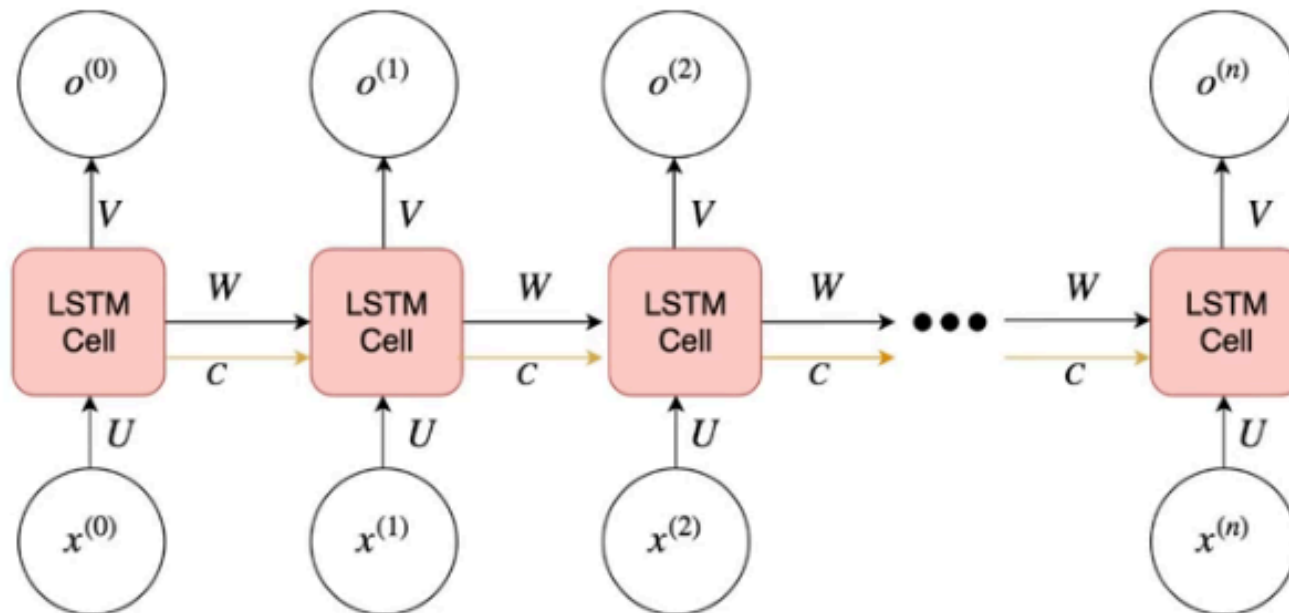




UNIVERSITAS
GADJAH MADA

LSTM

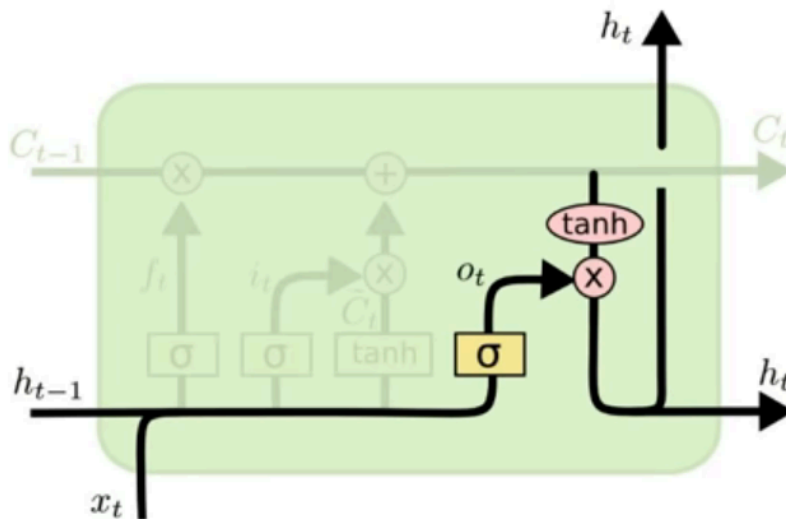
LSTM





UNIVERSITAS
GADJAH MADA

LSTM



Three Gates of LSTM Cell:

- **Input Gate** Is Cell Updated?

$$i^{(t)} = \sigma(W^i[h^{(t-1)}, x^{(t)}] + b^i)$$

- **Forget Gate** Is memory set to 0?

$$f^{(t)} = \sigma(W^f[h^{(t-1)}, x^{(t)}] + b^f)$$

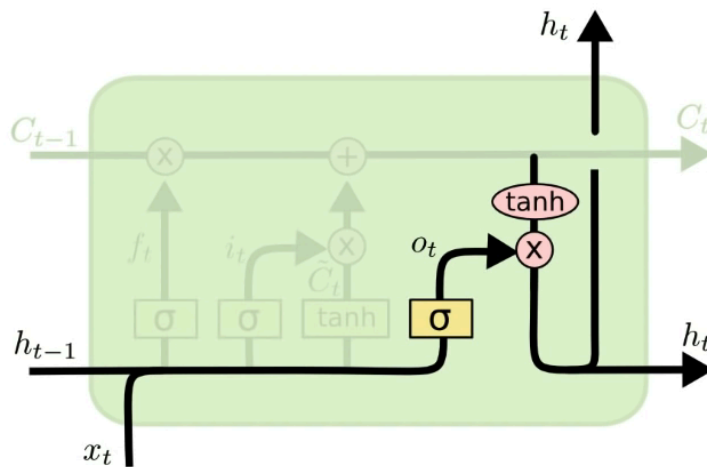
- **Output Gate** Is current info visible?

$$o^{(t)} = \sigma(W^o[h^{(t-1)}, x^{(t)}] + b^o)$$



UNIVERSITAS
GADJAH MADA

LSTM



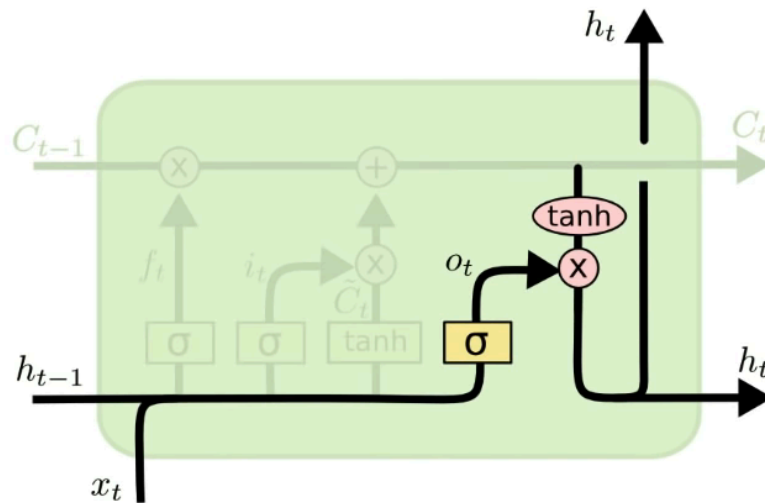
$$\bar{c}^{(t)} = \tanh(W^c[h^{(t-1)}, x^{(t)}] + b^c)$$

\tanh distributes gradients. Hence prevents vanishing/exploding



UNIVERSITAS
GADJAH MADA

LSTM



$$\bar{C}^{(t)} = \tanh(W^C[h^{(t-1)}, x^{(t)}] + b^C)$$

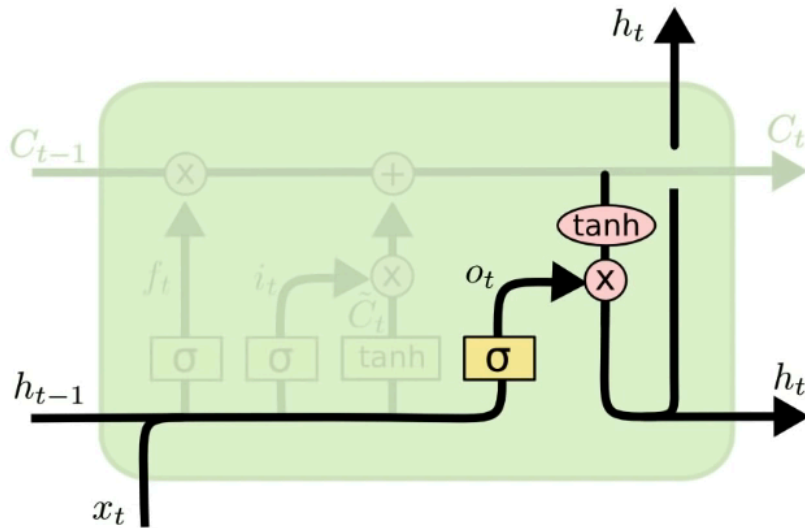
$$C^{(t)} = f^{(t)}C^{(t-1)} + i^{(t)}\bar{C}^{(t)}$$

$$h^{(t)} = \tanh(C^{(t)}) \times o^{(t)}$$



UNIVERSITAS
GADJAH MADA

LSTM



$$\bar{c}^{(t)} = \tanh(W^c[h^{(t-1)}, x^{(t)}] + b^c)$$

$$C^{(t)} = f^{(t)}C^{(t-1)} + i^{(t)}\bar{c}^{(t)}$$

$$h^{(t)} = \tanh(C^{(t)}) \times o^{(t)}$$

LSTM Parameters: $b^i, W^i, b^f, W^f, b^c, W^c, b^o, W^o$



UNIVERSITAS
GADJAH MADA

Metode LSTM

Gambar 1 Persamaan (2), h_{t-1} dan x_t akan melewati gerbang Sigmoid pada (1). Langkah pertama LSTM memutuskan informasi apa yang akan dihapus dari *cell state*. Keputusan ini dibuat oleh sigmoid *layer* yang bernama "*forget gate layer*". *Forget gate layer* akan memproses h_{t-1} dan x_t sebagai input, dan menghasilkan *output* berupa angka 0 atau 1 pada *cell state* C_{t-1} .

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

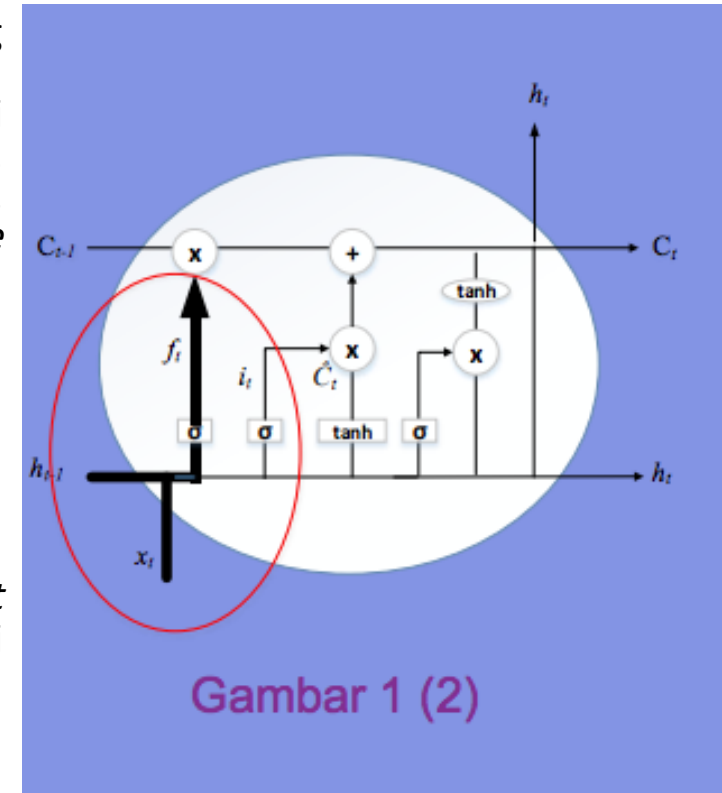
Dimana: x =data input, e =konstanta matematika (2,718281828459045235360287471352)

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2)$$

Dimana: f_t =*forget gate*, σ =fungsi sigmoid, W_f =nilai *weight* untuk *forget gate*, h_{t-1} =nilai *output* sebelum orde ke t , x_t =nilai input pada orde ke t , b_f =nilai bias pada *forget gate*.

Nilai *weight* diuraikan pada persamaan (3)

$$W = (-1/d^{1/2}), 1/d^{1/2} \quad (3)$$





Metode LSTM

Langkah selanjutnya adalah memutuskan informasi baru apa yang akan di simpan di *cell state* (4). Ada dua bagian. Pertama, lapisan sigmoid yang disebut “*input gate layer*” menentukan nilai mana yang akan diperbaharui. Selanjutnya, lapisan tanh membuat vektor nilai baru, \hat{C}_t , yang dapat ditambahkan ke *cell state* pada (6). Pada langkah selanjutnya, keduanya digabungkan untuk pembaharuan ke *cell state*.

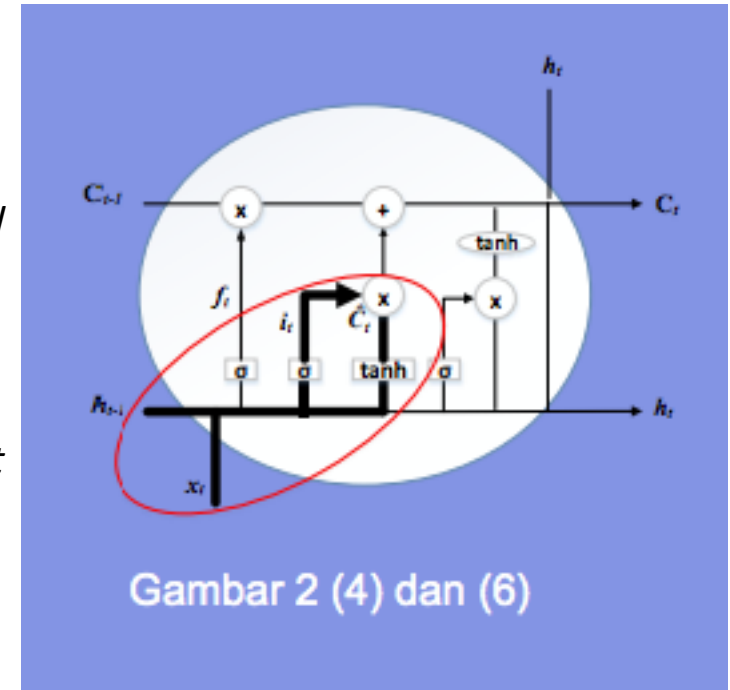
$$i_t = \sigma \cdot (W_i [h_{t-1}, x_t] + b_i) \quad (4)$$

Dimana: i_t = *input gate*, σ = fungsi sigmoid, W_i = nilai *weight* untuk *input gate*, h_{t-1} = nilai output sebelum orde ke t , x_t = nilai input pada orde ke t , b_i = nilai bias pada *input gate*

$$\tanh(x) = 2\sigma(2x) - 1 \quad (5)$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$

Dimana: \hat{C}_t = adalah nilai baru yang ditambahkan ke *cell state*, \tanh = fungsi tan hiperbolik, W_c = nilai *weight* untuk *cell state*, h_{t-1} = nilai output sebelum orde ke t , x_t = nilai input pada orde ke t , b_c = nilai bias untuk *cell state*



Gambar 2 (4) dan (6)



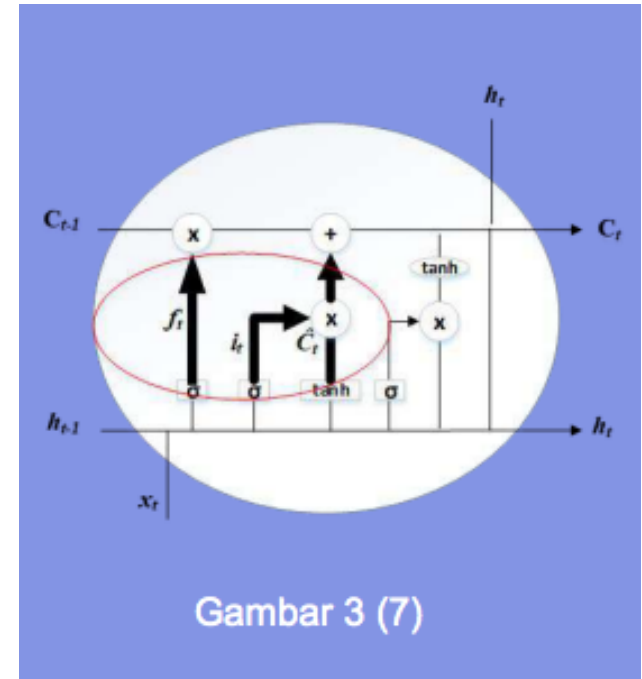
UNIVERSITAS
GADJAH MADA

Metode LSTM

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t \quad (7)$$

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.

Saatnya untuk memperbarui cell state lama (7), C_{t-1} , ke cell state baru C_t . Dengan mengalikan keadaan state lama dengan f_t , diputuskan untuk melupakan *forget gate layer*. Lalu tambahkan $i_t \times \hat{C}_t$. Ini adalah nilai baru, diskalakan dengan berapa banyak untuk diputuskan perbaharuan setiap cell state.



Gambar 3 (7)



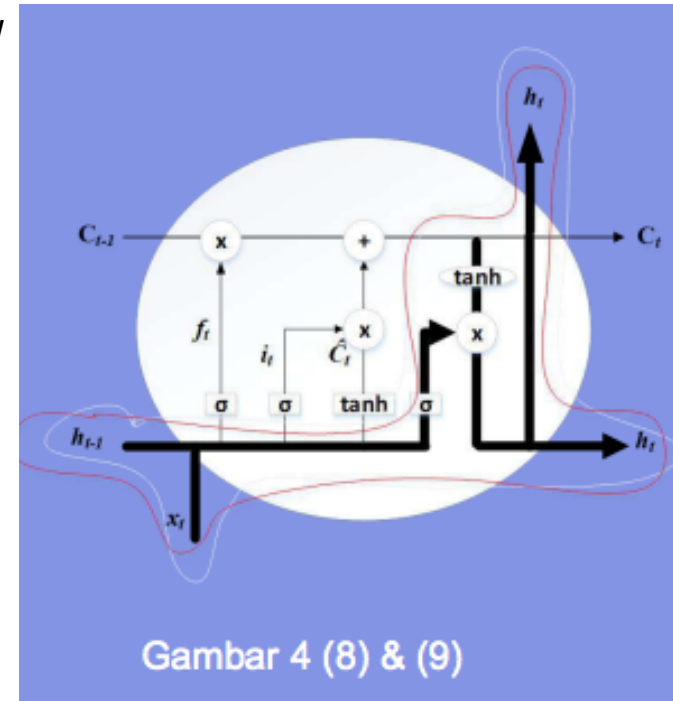
Metode LSTM

Langkah terakhir dari LSTM (8), output harus sesuai dengan *cell state* terlebih dahulu. Pertama, menjalankan lapisan sigmoid yang menentukan bagian *cell state* mana yang menjadi *output* (8). Kemudian, menempatkan *cell state* melalui tanh (untuk mendorong nilai antara -1 dan 1) dan mengalikannya dengan output dari gerbang sigmoid, sehingga hanya mengeluarkan bagian-bagian yang diputuskan (9).

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = O_t \times \tanh(C_t) \quad (9)$$

Dimana: O_t = output gate, σ = fungsi sigmoid, W_o = nilai weight untuk output gate, h_{t-1} = nilai output sebelum orde ke t, x_t = nilai output pada orde ke t, b_o = nilai bias pada output gate, h_t = nilai output orde t, O_t = output gate, \tanh = fungsi tanh, C_t = *cell state*.





Root Mean Square Error

Rumus RMSE menurut (Montgomery *et al*, 2008) dapat dilihat pada (10)

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (x_t - f_t)^2}{n}} \quad (10)$$

Dimana x_t adalah nilai aktual pada waktu ke- t , f_t adalah nilai prediksi pada waktu ke- t , n adalah jumlah data yang diprediksi.



UNIVERSITAS
GADJAH MADA

Augmented Dicky Ad Fuller Test

Augmented Dicky Ad Fuller Test (David Dickey & Wayne Fuller, 1979) adalah jenis uji statistik yang disebut Unit root test. Intuisi unit root test adalah menentukan seberapa kuat deret waktu ditentukan oleh tren.

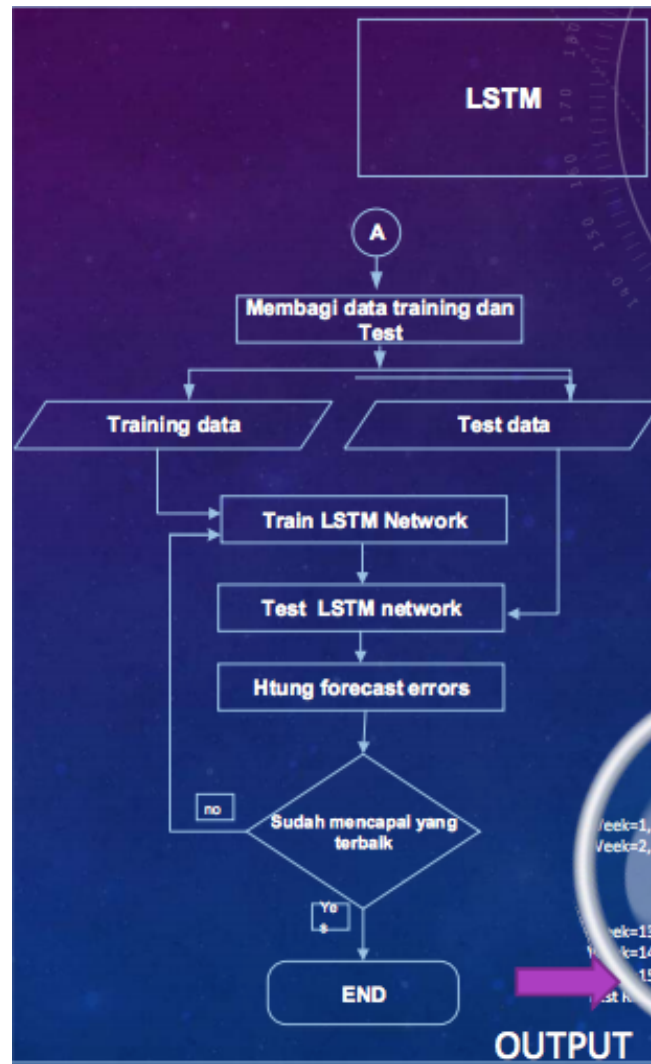
$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \dots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t \quad (11)$$

Menguji hipotesis nol yang menyatakan bahwa unit root ada dalam model autoregresif. Hipotesis alternatif berbeda tergantung pada versi tes yang digunakan, tetapi biasanya stasioneritas atau tren-stasioneritas.



UNIVERSITAS
GADJAH MADA

Diagram Model





UNIVERSITAS
GADJAH MADA

Perangkat Keras & Lunak

Perangkat keras yang digunakan dalam melakukan prediksi harga saham dalam slide ini adalah komputer personal dengan spesifikasi sebagai berikut:

- Prosesor: MacBook Pro (13-inch, Late 2011), 2.4 GHz Intel Core i5
- Memory: 8 GB 1333 MHz DDR3

Perangkat lunak yang digunakan adalah sebagai berikut:

- Sistem Operasi: macOS High Sierra version 10.13.6
- Anaconda 3, Jupyter Notebook 6.00, Bahasa Pemrograman Python 3.7, *Package* Keras, dan Tensorflow



UNIVERSITAS
GADJAH MADA

Pengumpulan Data

Dataset yang digunakan adalah dataset harga saham APPLE. Dataset dapat di download di <https://finance.yahoo.com> atau:

Di sini



UNIVERSITAS
GADJAH MADA

Practice

Pada praktik berikut ini, akan diprediksi harga penutupan (closing price) perusahaan APPLE.

```
#import the libraries
import math
import pandas_datareader as web
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

```
#Get the stock quote
df=web.DataReader('AAPL', data_source='yahoo', start='2012-01-01', end='201
#show the data
df
```



UNIVERSITAS
GADJAH MADA

Practice

	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	14.732143	14.607143	14.621429	14.686786	302220800.0	12.650659
2012-01-04	14.810000	14.617143	14.642857	14.765714	260022000.0	12.718646
2012-01-05	14.948214	14.738214	14.819643	14.929643	271269600.0	12.859850
2012-01-06	15.098214	14.972143	14.991786	15.085714	318292800.0	12.994284
2012-01-09	15.276786	15.048214	15.196429	15.061786	394024400.0	12.973674
2012-01-10	15.214286	15.053571	15.211071	15.115714	258196400.0	13.020121
2012-01-11	15.101786	14.975357	15.095714	15.091071	215084800.0	12.998899
2012-01-12	15.103571	14.955357	15.081429	15.049643	212587200.0	12.963211
2012-01-13	15.016071	14.952143	14.989286	14.993214	226021600.0	12.914607
2012-01-17	15.213929	15.105714	15.150000	15.167857	242897200.0	13.065036
2012-01-18	15.338214	15.225000	15.248571	15.325357	276791200.0	13.200701
2012-01-19	15.406071	15.232500	15.362500	15.276786	261738400.0	13.158866
2012-01-22	15.007357	14.991071	15.007500	15.010714	140071400.0	12.999999



UNIVERSITAS
GADJAH MADA

Practice

```
# Get the number of rows and columns in the data set  
df.shape
```

```
(2003, 6)
```

```
#Visualize the closing price history  
plt.figure(figsize=(16,8))  
plt.title('Close Price History')  
plt.plot(df['Close'])  
plt.xlabel('Data', fontsize=18)  
plt.ylabel('Close Price in USD ($)', fontsize=18)  
plt.show()
```



UNIVERSITAS
GADJAH MADA

Practice





UNIVERSITAS
GADJAH MADA

Practice

```
# Create a new dataframe with only the 'Close' column
data=df.filter(['Close'])
# Convert the dataframe to a numpy array
dataset = data.values
# Get the number of rows to train the model on
training_data_len = math.ceil(len(dataset)* .8)

training_data_len
```

1603

```
# Scale the data
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)

scaled_data
```

```
array([[0.01316509],
       [0.01457063],
       [0.01748985],
       ...,
       [0.97658263],
```



UNIVERSITAS
GADJAH MADA

Practice

```
#Create the training dataset  
#Create the scaled training dataset  
train_data = scaled_data[0:training_data_len, :]  
#Split the data into x_train and y_train data sets  
x_train = []  
y_train = []
```

```
for i in range(60, len(train_data)):  
    x_train.append(train_data[i-60 : i, 0])  
    y_train.append(train_data[i, 0])  
    if i<=61:  
        print(x_train)  
        print(y_train)  
        print()
```

```
[array([0.01316509, 0.01457063, 0.01748985, 0.02026915, 0.01984303,  
        0.02080338, 0.02036454, 0.01962679, 0.01862191, 0.02173194,  
        0.02453668, 0.02367172, 0.01893355, 0.02345548, 0.01900352,  
        0.03569838, 0.03440732, 0.0360927 , 0.03973694, 0.04194384,  
        0.0417594 , 0.0410789 , 0.04397903, 0.04670744, 0.04979839,  
        0.05479095, 0.0652785 , 0.06543749, 0.07127594, 0.07563885,  
        0.06814049, 0.07102789, 0.07097066, 0.07906688, 0.07791571,  
        0.08004628, 0.08387497, 0.08600558, 0.09214292, 0.09661394,  
        0.09790501, 0.09835659, 0.09071194, 0.08886753, 0.08914103,
```



UNIVERSITAS
GADJAH MADA

Practice

```
# Convert the x_train and y_train to numpy arrays
x_train, y_train = np.array(x_train), np.array(y_train)
```

```
#Reshape the data
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_train.shape
```

```
(1543, 60, 1)
```

```
#Build the LSTM model
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape= (x_train.shape[1], 1)
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```



UNIVERSITAS
GADJAH MADA

Practice

```
# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
#Train the model
model.fit(x_train, y_train, batch_size=1, epochs=1)
```

```
WARNING: Logging before flag parsing goes to stderr.
W0407 17:37:53.341605 140736055112576 deprecation_wrapper.py:119] From /Users/ivanluthfi/anaconda2/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
```

```
Epoch 1/1
1543/1543 [=====] - 384s 249ms/step - loss: 5.99
69e-04
```

```
<keras.callbacks.callbacks.History at 0x1a3a162208>
```



Practice

```
# Create the testing dataset
# Create a new array containing scaled values from index 1543 to 2003
test_data=scaled_data[training_data_len - 60: 2003]
#Create the data sets x_test and y_test
x_test=[]
y_test=dataset[training_data_len:, :]
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60: i, 0])
```

```
#Convert the data to a numpy array
x_test = np.array(x_test)
```

```
#Reshape the data
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

```
# Get the models predicted price values
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
```

```
# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean( predictions - y_test)**2)
rmse
```

```
0.09389039039611817
```




UNIVERSITAS
GADJAH MADA

Practice

```
# Plot
train = data[: training_data_len]
valid = data[training_data_len:]
valid['Predictions'] = predictions
#Visualize the data
plt.figure(figsize=(16, 8))
plt.title('Model')
plt.xlabel('Data', fontsize=18)
plt.ylabel('IDR', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='lower right')
plt.show()
```

```
/Users/ivanluthfi/anaconda2/lib/python3.6/site-packages/ipykernel_launcher
r.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
after removing the cwd from sys.path.



UNIVERSITAS
GADJAH MADA

Practice





UNIVERSITAS
GADJAH MADA

Practice

```
#Show the valid and predicted prices  
valid
```

	Close	Predictions
Date		
2018-05-17	46.747501	47.273289
2018-05-18	46.577499	47.225491
2018-05-21	46.907501	47.129574
2018-05-22	46.790001	47.079659
2018-05-23	47.090000	47.040680
2018-05-24	47.037498	47.063320
2018-05-25	47.145000	47.104683
2018-05-29	46.974998	47.169304
2018-05-30	46.875000	47.211533
2018-05-31	46.717499	47.226154
2018-06-01	47.560001	47.205418
2018-06-04	47.957500	47.306488
2018-06-05	48.327499	47.507278
2018-06-06	48.494999	47.780216



UNIVERSITAS
GADJAH MADA

Kesimpulan

- Secara keseluruhan LSTM mempunyai potensi yang baik bila digunakan untuk memprediksi data time series yang terbatas.



UNIVERSITAS
GADJAH MADA

Tugas

- Prediksi salah satu saham pada index LQ45 menggunakan LSTM.



UNIVERSITAS
GADJAH MADA

Terima kasih