# BAB II KAJIAN PUSTAKA

### 2.1 Pengertian Saham

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

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

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

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

#### 2.1.2 Harga Penutupan

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

#### 2.1.3 Prediksi Saham

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

## 2.2 *Recurrent neural network* (RNN)



Gambar 2.1 Arsitektur pada RNN

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

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

Gambar 2.2. Hidden layer pada RNN

……………………...(2.1)

………………………………….(2.2)

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

Gambar 2.3 Memori pada LSTM

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

* ***Input Layer***

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

* ***Hidden Layer***

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

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

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

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

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

…………………………………………. (2.3)

…………………………………………….(2.4)

* 1. ***Input gate***

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

1. Proses input asli dari perhitungan *output* sebelumnya waktu () dan *input* baru waktu , diproses dengan fungsi sigmoid. Di *paper* disebut (lapisan sigmoid input) [7].

………………………………….……… (2.5)

…………………………………………… (2.6)

1. Proses memperhitumgakan *output* sebelumnya dan *input* baru diproses dengan fungsi yang berbeda yaitu tanh. Di*paper* disebut (lapisan kandidat) [7].

……………………………………..... (2.7)

…………………………………..…(2.8)

Dari proses perkalian dari kedua hasil tersebut menghasilkan vektor baru dan disimpan di *memory cell* sebagai calon masukan , calon masukan tersebut akan dilanjutkan ke lapisan pernghubung ( antara hasil , calon masukan , dan *state* penghubung sebelumnya () diproses dengan rumus [7]. :

.……………………………………… (2.9)

* 1. ***Output gate***

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

1. Perhitungan *output* sebelumnya waktu () dan *input* baru waktu , diproses dengan fungsi sigmoid.

………………………………………..(2.10)

………………………………………… (2.11)

1. Proses mengontrolnya dua *state* dari *state* penghubung , dan *state* output diproses menghasilkan *block output* () dengan fungsi tanh [7] :

………………………………………… (2.12)

* ***Output Layer***

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

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

## 2.3 *Long Short Term Memory* *(LSTM)*

*Long Short Term Memory Networks* (LSTM) merupakan salah satu metode dari *Recurrent Neural Network* (RNN). LSTM diajukan oleh Sepp Hochreiter dan Jurgen Schimidhuber pada tahun 1927. LSTM di *design* untuk menghindari masalah *long term dependency* yang ada pada RNN (Grave, 2014) [4]. Didalam *hidden layer* LSTM yang saling berhubungan dan saling berkaitan, terdapat proses dalam menyeleksi suatu informasi untuk diingat dalam jangka waktu pendek maupun dalam jangka waktu panjang, setiap proses dalam satu urutan akan dijelaskan sebagai berikut :

+ ++ ++

+ ++ ++

+ ++ ++

+ ++ ++

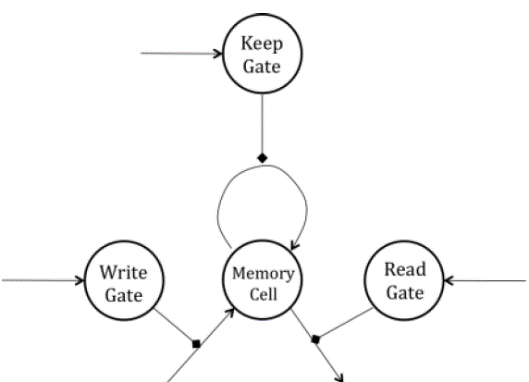
Gambar 2.4 Arsitektur LSTM

1. Status sel

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

Gambar 2.5 Status sel LSTM

1. Gerbang

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

Gambar 2.6 Gerbang LSTM

1. Penjagaan gerbang *(Keep gate)*

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

+ ++ ++

+ ++ ++

+ ++ ++

+ ++ ++

‘

Gambar 2.7 Penjagaan gerbang (Keep gate) LSTM

1. Menambahkan informasi baru *(write gate)*

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

+ ++ ++

+ ++ ++

+ ++ ++

+ ++ ++

Gambar 2.8 Menambahkan informasi baru (write gate) LSTM

1. Update *cell state* (

Proses melupakan hal-hal yang sudah diseleksi untuk dilupakan sebelumnya dan proses menambahkan informasi yang sudah ditentukan sebelumnya Secara rumus dapat dituliskan seperti persamaan (2.9)

+ ++ ++

+ ++ ++

+ ++ ++

+ ++ ++

Gambar 2.9 Update cell state LSTM

1. Membaca keputusan *(Output)*

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

+ ++ ++

+ ++ ++

+ ++ ++

+ ++ ++

Gambar 2.10 Membaca keputusan (Output) LSTM

Lapisan gerbang output dijabarkan dengan persamaan (2.11)

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

### Algoritma pelatihan *Long Short Term Memory Networks*

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

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

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

Berikut langkah-langkah pelatihannya adalah :

**Langkah 0 :**

* 1. Persiapkan data normalisai
  2. Inisialisasi dimensi *input* dan *output*
  3. Tetapkan *Epcohs* (satu presentasi lengkap dari proses *forward* sampai *Backward* untuk pemrosesan data yang akan dipelajari ke mesin pembelajaran)
  4. Inisialisasi bobot (bobot awal dengan nilai random yang cukup kecil)
  5. Tetapkan maksimum target *error* dan *learning rate*
  6. Tetapkan panjang urutan (*sequence length)*

**Langkah 1 :**

1. Pengabungan antara input data dan block output dari langkah sebelumnya:

………………………………………… …….…………….(2.13)

: Hasil penggabungan *input*

: Hasil block output sebelumya

: Input data yang akan diolah

1. Block input dengan mempersiapkan input :

……………………………………………………... (2.14)

Catatan :

* Baris jika simpul 1 maka 4 x 1 = 4
* Kolom jika *size* input 2 maka 2 + simpul = 3
* Jadi ordo matrik adalah (4,3)

……………………………………………………..(2.15)

………………………………………………(2.16)

…………………………………………………………………(2.17)

Keterangan :

: Hasil *input (hidden layer)* terdapat 4 baris untuk *input*

: *input (hidden layer)* pada gerbang *forget* seperti persamaan *(2.3)*

: *input (hidden layer)* pada gerbang *input* seperti persamaan *(2.5)*

: *input (hidden layer)* pada *candidate state* seperti persamaan *(2.7)*

: *input (hidden layer)* pada gerbang *output* seperti persamaan *(2.10)*

: Bobot (weight) yang menghubungkan input layer

:Data informasi

### Feedforward

**Langkah 2 :**

1. Gerbang lupa *(forget gate)*:

………………………………….…………………….(2.3)

Keterangan :

|  |  |
| --- | --- |
|  | : Hasil perhitungan gerbang lupa *(forget gate*) |
|  | :*Block Input forget gate* |
|  | : Nilai *input* dari *hidden output* |
|  | : Bias |
|  | : Logistic sigmoid () |

1. Gerbang pembuatan vektor barudandisimpan di *memory cell*:
2. ……………………………………………...(2.6)
3. ………………………………………… (2.8)

Keterangan :

: Hasil perhitungan gerbang *input* *(input gate)*

: Proses tanh *(candidate gate)*

Fungsi tanh digunakan membuat vektor baru

1. Keadaanpenghubung *(state cell) :*

………………………………………………….. (2.9)

Keterangan :

: Hasil perhitungan keadaan penghubung *(connecting state)*

: Keadaan penghubung sebelumnya

1. Gerbang *output* *(output gate) :*

……………………………………………………(2.11)

Keterangan :

: Hasil perhitungan gerbang *output (output gate)*

1. *Block output*

………………………………………………………………………………(2.12)

Keterangan :

: Hasil perhitungan *output* dan hasil prediksi

### Backpropagation Through Time (BPTT)

**Langkah 3 :**

()

Keterangan :

: Harga real (data asli)

: Harga prediksi (data prediksi)

Pada proses forward diberikan dan ditemukan

1. Hitung informasi *derivative* gerbang output :

………………………………………………. (2.18)

Keterangan :

:Menghitung informasi *derivative* gerbang output

: Informasi *derivative* output block

:keadaan penghubung *(connecting state)*

1. Hitung informasi *derivative* keadaan penghubung *(cell state)* :

………………………………..(2.19)

Keterangan :

: Menghitung informasi *derivative* keadaan penghubung

: Gerbang *Output*

**Langkah 4 :**

Pada proses forward diberikan dan ditemukan

1. Hitung informasi derivative gerbang input (*input gate*) :

………………………………………………………..(2.20)

Keterangan :

:Menghitung informasi *derivative input gate*

: Informasi *derivative* keadaan penghubung

: *candidate gate*

1. Hitung informasi *derivative* gerbang *candidate connecting state*

...…………………………………………………….. (2.21)

Keterangan :

:Informasi *derivative* *candidate connecting state*

: Informasi *derivative* Keadaan penghubung

: *input gate*

1. Hitung informasi *derivative* gerbang *forget gate*

………………………………………… …………...(2.22)

Keterangan :

:Informasi *derivative* *forget gate*

: Informasi *derivative* keadaan penghubung

: Keadaan penghubung sebelumnya

1. Hitung informasi derivative Keadaan penghubung sebelumnya

……………………………………………………,,, (2.23)

Keterangan :

:Informasi *derivative* keadaan penghubung sebelumnya

: Informasi *derivative* keadaan penghubung

: Gerbang *forget gate*

**Langkah 5 :**

Pada proses forward diberikanmaka pada fase backward didapat = .

1. Hitung *derivative* *block input* *(candidate gate)* :

...…………………………………….(2.24)

Keterangan :

:Informasi *derivative* *input block* *candidate gate*

: Informasi *derivative* *candidate gate*

: *block input* *candidate gate*

1. Hitung *derivative* *block input* *(input gate)*:

………………………………………..…(2.25)

Keterangan :

:Informasi *derivative* *input block* *input gate*

: Informasi *derivative* *input gate*

: *input gate*

1. Hitung *derivative* *block input (forget gate)* :

………………………………………. (2.26)

Keterangan :

:Informasi *derivative* *input block* *forget gate*

: Informasi *derivative* *forget gate*

: *forget gate*

1. Hitung *derivative* *block input (output gate)* :

……………………………………….. (2.27)

:Informasi *derivative* *input block* *output gate*

: Informasi *derivative* *output gate*

: *output gate*

1. Penggabungan dari menjadi satu untuk menghasilkan dari *forward* terdapat maka perlu pengembalian :

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

…………………………………….. (2.28)

………………………………………………………... (2.29)

### Mencari Bobot dari proses backpropagation urutan (sequence)

**Langkah 6 :**

Dari tahapdidapat persamaan (2.15) dan diberikan dari langkah sebelumnya maka dicari

1. Persamaan (2.29) didapat untuk menghitung

……………………………………………………….. (2.30)

………………………………….. (2.31)

1. Dari persamaan (2.31), didapat untuk menghitung *derivative block output*

] ………………………………………………… (2.32)

1. Mencari *derivative* pada bobot yang akan menjadi model dalam prediksi selanjutnya

………………………………………………………..(2.33)

…….(2.34)

1. Menghitung Derivatif Bobot

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

(2.34)

…………………………………………………………...(2.35)

……………………………….(2.36)

Keterangan :

= *derivative* bobot

dihitung sampai t =1

### Optimasi untuk update bobot Gradient Descent

**Langkah 7 :** *Gradient descent weight optimization*

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

Dapat dituliskan dengan persamaan :

………………………. (2.37)

Keterangan :

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

bobot ini akan digunakan pemodelan.

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

**Langkah 8 :** Hitung total *error*

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

…………………………………………… (2.38)

Keterangan :

*MSE* : Merupakan total *error*

: Merupakan target *output* ke-i

: Merupakan *output* dari pelatihan ke-i

Fase tersebut diulang hingga kondisi *error* terpenuhi.

## 2.4 Penelitian Terkait

Table 1. Tabel Penelitian Terkait

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

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

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