

# *Prediksi Harga Saham menggunakan Metode Recurrent Neural Network*

M Abdul Dwiyanto Suyudi, Esmeralda C. Djamil, Asri Maspupah  
Jurusan Informatika, Fakultas Sains dan Informatika  
Universitas Jenderal Achmad Yani  
Cimahi, Indonesia  
Abdulsuyudi0701@gmail.com

**Abstrak**—Saham merupakan salah satu pilihan investasi yang menarik karena dapat diperoleh untung yang besar dibandingkan dengan usaha lainnya. Untuk meminimalkan resiko kerugian, diperlukan perhatian yang jeli terhadap pergerakan saham dan perkembangan pasar modal merupakan salah satu indikator yang perlu dipantau. Dengan teknologi pemrosesan prediksi dan pembelajaran mesin saat ini, identifikasi prediksi harga saham dapat dilakukan secara otomatis. Deep Learning merupakan salah satu bagian dari pembelajaran mesin, dan memiliki akurasi pengenalan yang tinggi dengan data yang sangat banyak. Penelitian ini menggunakan analisis *history* harga saham dalam suatu perusahaan, dan Recurrent Neural Network (RNN) untuk melakukan prediksi terhadap nilai saham dari *history* harga saham. Fitur yang diidentifikasi yaitu harga terendah, harga tertinggi, harga buka, harga tutup, volume, rata-rata harga, dan pergerakan. Prediksi tujuh fitur variabel dengan RNN menghasilkan akurasi sebesar 94% untuk data latih dan 55% untuk data uji. Akurasi diperoleh setelah pelatihan dengan menggunakan 1218 data.

**Kata kunci**— Saham, Recurrent Neural Network, Prediksi.

## I. PENDAHULUAN

Saham merupakan salah satu pilihan investasi yang menarik karena dapat diperoleh untung yang besar dibandingkan dengan usaha lainnya. Walaupun beresiko pula kerugian yang besar dalam waktu yang singkat. Untuk meminimalkan resiko kerugian, diperlukan perhatian yang jeli terhadap pergerakan saham [1]. Perkembangan harga saham dapat dilihat pada Indeks Harga Saham Gabungan (IHSG) yang memperlihatkan bahwa kenaikan harga saham merefleksikan antusiasnya pasar dan begitu pula sebaliknya [2].

Pergerakan harga saham sulit untuk ditebak akan arah kelajuannya. Dalam memprediksi dapat dilakukan dengan tiga faktor yang mempengaruhi, yaitu faktor teknikal, faktor fundamental, dan faktor sentiment [3]. Faktor teknikal merupakan pergerakan dengan cara mengamati harga pada masa lalu, faktor fundamental merupakan teknik analisis pendekatan secara bisnis yang terjadi, faktor sentiment merupakan pergerakan harga saham yang dipengaruhi oleh faktor-faktor bisnis, berita dan pelaku operasional bisnis.

Banyak metode dan cara yang dilakukan untuk memprediksi dalam hal jual beli saham. Dalam seharinya

diperoleh data *high price*, *low price* [4], *open price* [5], *close price*, *volume* dan perubahan setiap harinya. *High price* merupakan pencapaian harga tertinggi dalam satu hari. Sementara *low price* adalah pencapaian harga terendah dalam satu hari. *Open price* merupakan harga pembukaan dalam satu hari. *Close price* merupakan harga penutupan dalam satu hari. Volume merupakan jumlah bursa yang diperdagangkan dalam satu hari. Perubahan merupakan persentasi perubahan pergerakan harga dari waktu ke waktu.

Dewasa ini, perkembangan komputasi untuk pembelajaran mesin sangat pesat, yang berkembang salah satunya yaitu Deep Learning [6], mengingat perkembangan Graphics Processing Unit (GPU) yang sudah mendukung untuk melakukan pembelajaran data. Dengan menggunakan GPU proses pelatihan data menjadi 44 kali lebih cepat dibandingkan dengan menggunakan processor. Deep Learning sangat memungkinkan melakukan pembelajaran dengan lapisan yang lebih kompleks agar mendapatkan akurasi yang tinggi tetapi dalam Deep Learning dapat belajar lebih efisien karena dapat belajar dalam Jaringan Syaraf Tiruan yang lebih dalam atau lapisan tersembunyi yang lebih banyak. Dalam Deep Learning terdapat sebuah jaringan yang disebut Recurrent Neural Network (RNN) yang dapat digunakan untuk mengolah data *Time Series*.

Beberapa penelitian terdahulu untuk prediksi harga saham di antaranya menggunakan metode algoritma Genetika dan menggunakan model Regresi. prediksi saham ditinjau dari empat fitur yaitu harga buka, harga tutup, harga paling tinggi, dan harga paling rendah mendapatkan tingkat akurasi sebesar 73,78% hanya kurang realistis dalam memprediksi harga saham sehingga akurasi yang didapatkan tidak sesuai dengan nilai harga saham sebenarnya [7]. Sementara penelitian lain menggunakan lima fitur yang disertai penambahan fitur volume menggunakan metode algoritma Koordinate Sub-Mode mendapatkan tingkat akurasi sebesar 62,29% hanya dibutuhkan komputasi yang besar dan waktu yang lama [4]. Namun kelemahannya adalah keterbatasan data yang diproses sebanyak sepuluh data periode [7]. Prediksi harga saham dapat dilakukan dengan menggunakan metode Deep Convolutional Neural Network dengan menggunakan data pada sinyal CSV dengan matriks yang dikonvolusi untuk mendapatkan sejumlah fitur yang digunakan untuk pengenalan pola dan prediksi harga saham dengan hasil recall 70% [8].

Penelitian ini membangun sistem yang dapat memprediksi harga saham menggunakan faktor teknikal, dengan mengeluarkan perkiraan prediksi pada pergerakan saham beli atau jual. Variabel masukan yang digunakan adalah data harga paling tinggi, harga paling rendah, harga buka, harga tutup, rata-rata, volume dan perubahan. Sistem tersebut dibangun dari pembelajaran mesin yang dapat memprediksi dengan menggunakan data latih sebanyak 1218 data yang diambil dari suatu perusahaan menggunakan RNN dengan LSTM.

## II. FITUR DATASET HISTORY SAHAM

Dalam penelitian ini prediksi harga saham dilakukan berdasarkan tujuh fitur yaitu data harga paling tinggi, harga paling rendah, harga buka, harga tutup, volume [5], rata-rata dan perubahan yang diperoleh dalam suatu perusahaan. Berikut contoh data yang didapat pada Tabel 1.

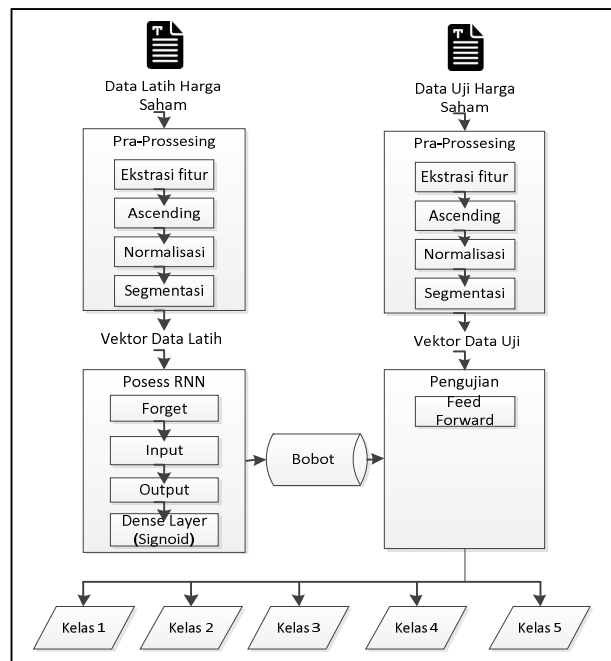
TABEL I. HISTORY DATA SAHAM

Tanggal	Buka	Tertinggi	Terendah	Tutup	Volume	Perubahan
16/3/2012	6750	6750	6350	6750	19,31M	0%
19/3/2012	6750	7100	6550	6750	14,58M	0%
20/3/2012	6800	6900	6600	6800	31,15M	-1,47%
21/3/2012	6750	6900	6700	6750	16,87M	0,75%
22/3/2012	6500	6900	6500	6500	29,31M	0,74%
.	.	.	.	.	.	.
.	.	.	.	.	.	.
13/2/2018	4900	4900	4850	4900	108,24M	2,11%
14/2/2018	4900	4900	4900	4900	67,67M	-0,52%
15/2/2018	4975	5050	4925	4975	52,89M	-0,52%

Prediksi ini dilakukan untuk harga nilai kedepannya untuk mengantisipasi kerugian yang didapat oleh pelaku pemain saham. Oleh karena itu diperlukan prediksi harga saham untuk para pemain saham tidak salah mengambil langkah untuk menjual atau membeli saham dalam suatu perusahaan, karena data yang diperoleh dari suatu perusahaan berupa data harian sehingga akan dapat pergerakan nilai saham secara rinci.

## III. METODE

Prediksi akan diproses dengan analisis struktur untuk mengidentifikasi fitur data harga paling tinggi, harga paling rendah, harga buka, harga tutup, rata-rata, volume dan perubahan setelah itu melalui tahap *pra-proses* masuk ke dalam ekstraksi fitur, *ascending*, normalisasi dan segmentasi. Setelah data tersebut dilakukan *pra-proses* dilanjutkan dengan memasukan proses Recurrent Neural Network (RNN) [3] data tersebut yang telah terstruktur dimasukan ke dalam *forget gate*, *input gate*, *output gate*, dan *dense layer* (Sigmoid) untuk menentukan bobot yang dihasilkan dari metode RNN tersebut. Selanjutnya melakukan pengujian dengan menggunakan *feed forward* untuk mendapatkan nilai akhir, setelah mendapatkan nilai akhir maka dimasukan ke dalam kelas yang telah dikasih rentang sebanyak lima kelas. Proses tersebut dapat dilihat pada Gambar 1.



Gambar 1. Prediksi Harga Saham

### A. Pra-prosesing

*Pra-prosesing* mengubah data yang belum terstruktur menjadi data yang terstruktur dengan kebutuhan prosesnya, maka akan dibagi menjadi 4 bagian *pra-prosesing* yaitu ekstraksi fitur, *ascending*, normalisasi, dan segmentasi.

#### 1) Ekstraksi Fitur

Ekstraksi fitur untuk menambahkan fitur baru yaitu fitur rata-rata.

Nilai rata-rata didapatkan dari:

$$\text{Rata - Rata} = \frac{\text{Buka} + \text{Tertinggi} + \text{Terendah} + \text{Tutup}}{4} \quad (1)$$

Data pengambilan rata-rata dapat dilihat pada Tabel 2.

TABEL 2. HISTORY DATA SAHAM RATA-RATA.

Tanggal	Buka	Tertinggi	Terendah	Tutup	Volume	Perubahan	Rata-Rata
16/3/2012	6750	6750	6350	6750	19,31M	0%	6650
19/3/2012	6750	7100	6550	6750	14,58M	0%	6786,5
20/3/2012	6800	6900	6600	6800	31,15M	-1,47%	6775
21/3/2012	6750	6900	6700	6750	16,87M	0,75%	6775
22/3/2012	6500	6900	6500	6500	29,31M	0,74%	6600
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
13/2/2018	4900	4900	4850	4900	108,24M	2,11%	4887,5
14/2/2018	4900	4900	4900	4900	67,67M	-0,52%	4900
15/2/2018	4975	5050	4925	4975	52,89M	-0,52%	4981,25

#### 2) Ascending

*Ascending* dilakukan untuk mengurutkan data harga saham, *ascending* yang dibutuhkan untuk proses ini yaitu mengurutkan data saham dari yang terdahulu hingga yang terbaru.

### 3) Normalisasi

Normalisasi merupakan tahap kedua dari *pra-proses*. Pada tahap ini terjadi proses penskalaan, dimana data akan diubah dalam rentang 0 hingga 1. Hal ini dilakukan dikarenakan terdapat tiga jenis data. Data pertama yaitu data harga, yaitu harga buka, harga tutup, harga tertinggi, harga terendah, dan rata-rata. Sedangkan data kedua yaitu data volume, selain perbedaan data ada kemungkinan dimana data akan menjadi terlalu besar pada volume. Data ketiga yaitu perubahan nilai dengan persenan '%'. Dengan demikian normalisasi dilakukan untuk menyamaratakan data. Berikut keterangan Normalisasi Tabel 3.

TABEL 3. HISTORY DATA SAHAM NORMAL

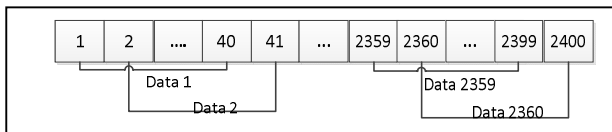
Tanggal	Buka	Terti- nggi	Terend- -ah	Tutup	Volume	Perub- -ahan	Rata- Rata
16/3/2012	0,75	0,75	0,46	0,75	0,15	0,00	0,90
19/3/2012	0,75	1	0,60	0,75	0,12	0,00	1
20/3/2012	0,84	0,92	0,69	0,84	0,19	-1,47%	0,92
21/3/2012	0,80	0,92	0,76	0,80	0,13	0,75%	0,92
22/3/2012	0,61	0,92	0,61	0,61	0,17	0,74%	0,89
.	.	.	.	.	.	.	.
13/2/2018	0,42	0,52	0,40	0,42	0,46	2,11%	0,44
14/2/2018	0,42	0,52	0,38	0,42	0,31	-0,52%	0,48
15/2/2018	0,45	0,45	0,45	0,45	0,28	-0,52%	0,46

Nilai Normalisasi didapat dari:

$$x_i = \frac{y - \text{valMin}()}{\text{valMax}() - \text{valMin}()} \quad (2)$$

### 4) Segmentasi

Segmentasi merupakan proses pemisahan dan pengelompokan data, dari data mentah menjadi data yang dibutuhkan sistem. Pada penelitian ini dijadikan panjang 40 hari. Berikut simulasi segmentasi pada Gambar 2:



Gambar 2. *Pra-proses* Segmentasi

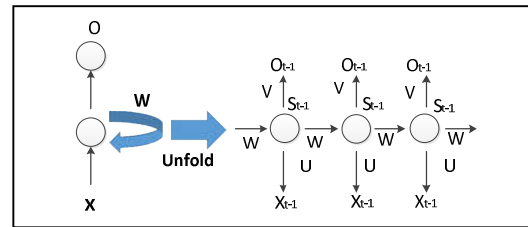
### B. Vektor Data Latih

Vektor data latih digunakan untuk mengubah nilai yang awalnya berbentuk *array* menjadi vektor, dikarenakan struktur RNN hanya bisa menerima masukan berbentuk *sekuensial*.

### C. Recurrent Neural Network

RNN merupakan bagian dari area *machine learning* yang berbasis ekstraksi fitur dari sebuah data secara lebih rinci. RNN ini pengembangan dari Jaringan Syaraf Tiruan (JST) dan arsitekturnya hampir sama dengan Multi Layer Perception (MLP) Cara kerja Deep Learning meniru cara kerja dari otak manusia sehingga dapat mengirimkan informasi dari sebuah *neuron* menuju *neuron* lain [9]. Recurrent Neural Network (RNN) memiliki variasi jenisnya seperti Long Short Term Memory[6]. LSTM ini digunakan untuk mengatasi masalah

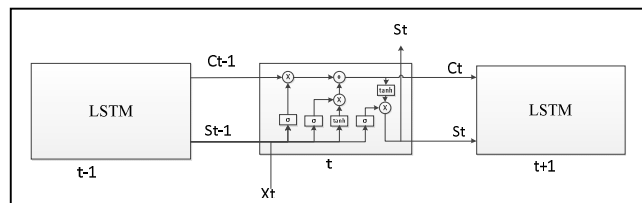
yang ada dalam RNN yaitu ingatan memori jangka pendek. Arsitektur RNN dapat di lihat pada Gambar 3.



Gambar 3. Arsitektur Recurrent neural Network

Long Short Term Memory (LSTM) jaringan yang diciptakan dengan tujuan mengatasi masalah *hidden layer*. Kunci dalam desain LSTM adalah untuk menggabungkan kontrol *non-linier*, data *dependent* ke dalam sel RNN [10], yang dapat dilatih untuk memastikan bahwa gradien fungsi tujuan dengan memperhatikan sinyal (kuantitas berbanding lurus dengan pembaruan parameter yang dihitung selama pelatihan oleh Gradient Descent) tidak menghilang.

LSTM digunakan untuk mengatasi *vanishing gradient* atau situasi dimana nilai *gradient* bernilai 0 atau mendekati 0 dengan mekanisme *gate* [11]. LSTM merupakan cara lain untuk menghitung *hidden state*. Memori dalam LSTM disebut dengan *cells* yang mengambil *input* dari *state* sebelumnya ( $h_{t-1}$ ) dan *input* saat ini ( $x_t$ ). Kumpulan *cells* tersebut memutuskan apa yang akan disimpan dalam memori dan kapan yang akan dihapus dari memori. LSTM menggabungkan *state* sebelumnya, memori saat ini, dan *input*. LSTM sangat efisien untuk merekam *long-term dependencies* terlihat.



Gambar 4. Arsitektur Long Short Term Memory

Dilihat dari Gambar 4, LSTM mempunyai 3 *gate*, yaitu *gate* yang pertama adalah *forget gate* untuk menentukan informasi yang akan dihilangkan dari *cell* menggunakan *sigmoid layer* seperti terlihat pada Persamaan 10 dengan fungsi aktivasi yang digunakan adalah Relu. *Gate* yang kedua adalah *input gate* yang mana dari *sigmoid layer* yang akan diperbarui dan *tanh layer* akan dibuat sebagai sebuah vektor dari nilai yang diperbarui [12]. *Gate* yang ketiga adalah *output gate* untuk memaparkan isi sel memori pada output LSTM.

LSTM diperlakukan sebagai kotak hitam yang diberikan *current input* dan *hidden state* sebelumnya untuk menghitung *hidden state* selanjutnya. Memori yang digunakan dalam LSTM disebut dengan *cells* yang mengambil *input* dari *state* sebelumnya ( $h_{t-1}$ ) dan *input* saat ini ( $x_t$ ). LSTM sangat cocok untuk pembelajaran *long-term dependencies* terlihat pada Persamaan 3, Persamaan 4, Persamaan 5, Persamaan 6, Persamaan 7, dan Persamaan 8.

$$i = \sigma(x_t U^i + s_{t-1} W^i) \quad (3)$$

$$f = \sigma(x_t U^f + s_{t-1} W^f) \quad (4)$$

$$o = \sigma(x_t U^o + s_{t-1} W^o) \quad (5)$$

$$g = \tanh(x_t U^g + s_{t-1} W^g) \quad (6)$$

$$c_t = c_{t-1} \circ f + g \circ i \quad (7)$$

$$s_t = \tanh(c_t) \circ o \quad (8)$$

Dimana  $i$ ,  $f$ ,  $o$  merupakan *gate input*, *forget*, *output*,  $g$  merupakan kandidat *hidden state*,  $c_t$  adalah unit memori internal yang merupakan kombinasi dengan memori sebelumnya,  $c_{t-1}$  adalah memori sebelumnya,  $s_t$  merupakan *hidden state* terbaru, lalu  $x_t$  adalah masukan pada setiap *time step* saat ini dan  $s_{t-1}$  adalah *hidden state* sebelumnya.

*Forget gate* digunakan untuk menentukan informasi yang akan dilupakan dari *cell* menggunakan sigmoid *layer*, dapat dilihat pada Persamaan 9 dengan fungsi aktivasi yang digunakan adalah Relu yang terlihat pada Persamaan 10.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (9)$$

$$\text{ReLU}(x) = \max(0, x) \quad (10)$$

*Input gate* yang berasal dari sigmoid *layer* yang akan diperbarui dan *tanh layer* akan dibuat sebagai sebuah vektor dari nilai yang diperbarui. Dapat dilihat pada Persamaan 11 dan 12.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (11)$$

$$c_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (12)$$

Kemudian *cell* dari Persamaan 10, Persamaan 11, dan Persamaan 12 akan diperbarui menggunakan Persamaan 13.

$$c_t = f_t * c_{t-1} + i_t * c_t \quad (13)$$

*Output gate* akan dikalkulasikan berdasarkan perbaruan *cell* dan sigmoid *layer* terlihat seperti Persamaan 14 dan 15.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (14)$$

$$h_t = o_t * \tanh(c_t) \quad (15)$$

Dimana  $\sigma$  merupakan fungsi aktivasi sigmoid dengan rentang (-1, 1) lalu  $\tanh$  merupakan fungsi aktivasi tangen dengan rentang (-1, 1) sedangkan  $W_f$ ,  $W_i$ ,  $W_c$ ,  $W_o$  adalah matriks bobot, untuk  $h_{t-1}$  adalah *hidden state* sebelumnya dan  $b_f$ ,  $b_i$ ,  $b_c$ ,  $b_o$  merupakan vektor bias.

#### IV. HASIL DAN DISKUSI

Data saham dari tahun 2012 hingga 2018 terdiri dari 1218 data. Data tersebut dibagi menjadi dua bagian yaitu 80% untuk data latih dan 20% untuk data uji. Pengujian dilakukan terdiri dari pengujian model optimasi dengan membandingkan dua model, pengujian jumlah data dan juga pengujian *learning rate*.

##### A. Membandingkan Antara Dua Model Optimasi

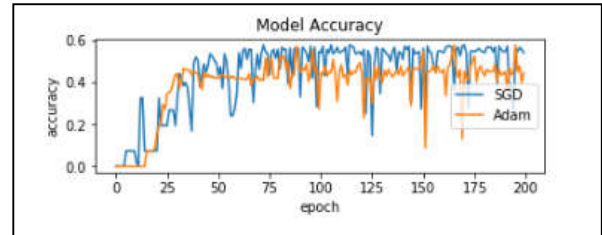
Penelitian ini menggunakan metode optimasi Stochastic Gradient Descent (SGD) dan Adaptive Moment Estimation (Adam) [13]. Metode ini merupakan suatu metode yang sering digunakan untuk model optimasi.

Sementara itu, kedua model optimasi menggunakan *learning rate* = 0,001 dan *epochs* sebanyak 200 untuk mendapatkan pelatihan yang lebih detail dari kedua metode optimasi tersebut. Hasil dari pengujian menggunakan SGD dan Adam menghasilkan *error* yang tidak terlalu signifikan dan keduanya hampir sama baiknya. Hasil pengujian dengan kedua metode optimasi tersebut dapat dilihat pada Tabel 3.

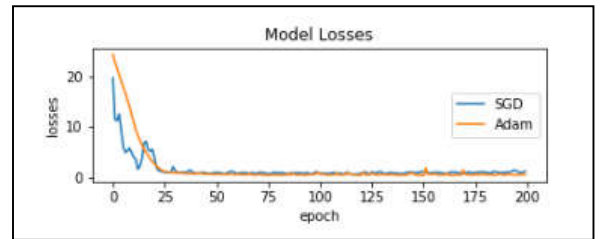
TABEL 4. HASIL UJI MODEL

No.	Optimasi	Data Latih		Data Uji	
		Loss	Akurasi	Loss	Akurasi
1.	SGD	0.1877	79.31%	0.4443	46.32%
2.	Adam	0.0563	94.16%	0.3871	55.26%

Metode SGD dan Adam memiliki kekurangannya dan kelebihan masing-masing, dari hasil penelitian ini metode optimasi SGD lebih unggul dibandingkan dengan Adam, dikarenakan memiliki tingkat *Loss* yang lebih rendah dan memiliki tingkat akurasi yang lebih tinggi. Grafik hasil pengujian SGD dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Model Akurasi SGD dan Adam



Gambar 6. Model Losses SGD dan Adam

Berdasarkan hasil pengujian penelitian model optimasi SGD dan Adam yang dapat dilihat pada Tabel 4 menunjukkan bahwa Adam memiliki keunggulan dalam model optimasi dengan mendapatkan nilai akurasi 94.16% untuk data latih dan 55,26% untuk data uji. Sedangkan untuk model optimasi Adam memiliki nilai akurasi 79,31% untuk data latih dan 46,32% untuk data uji. Grafik pengujian SGD dan Adam dijelaskan pada Gambar 5 untuk Model Accuracy dan Gambar 6 untuk Model Losses. Perbedaan SGD dengan Adam yaitu pengambilan data secara acak dan mengambil beberapa bagian dari data training.

##### B. Pengujian Learning Rate

Percobaan dilakukan dengan *learning rate* 0,001, 0,002, 0,050, 0,100, 0,200, 0,500 dan 0,900. Model optimasi Adam yang digunakan [14]. Hal ini dilakukan untuk mengetahui pengaruh ukuran *learning rate* terhadap proses pembelajaran dan juga pengujian data. Hasil yang diperoleh dari percobaan *learning rate* dapat dilihat pada Tabel 5.

TABEL 5 PENGUJIAN LEARNING RATE

No.	Learning Rate	Data Latih		Data Uji	
		Loss	Accuracy (%)	Loss	Accuracy (%)
1.	0.001	0.0563	94.16	0.3871	55.27
2.	0.002	0.0569	94.60	0.6232	57.89
3.	0.050	0.0597	94.86	1.1026	56.32
4.	0.100	0.0722	92.62	0.5559	54.89
5.	0.200	0.0790	91.44	0.9595	52.84
6.	0.500	0.0596	83.28	0.8454	53.84
7.	0.900	0.0673	91.47	0.7159	51.89

Pada tabel 5 dapat dilihat hasil akurasi data yang diperoleh berdasarkan *learning rate* yang bervariasi. Nilai akurasi tertinggi yaitu sebesar 57.89% untuk data uji dengan nilai *learning rate* 0.002, kemudian nilai akurasi untuk data latih tertinggi sebesar 94,86% dengan *learning rate* 0.050. Pada hasil *learning rate* 0.001 hingga 0.900 selisih perbedaan tingkat akurasi data uji dan data latih tidak terlalu signifikan. Hal ini dikarenakan ketika pembelajaran suatu mesin terlalu besar, lalu nilai *gradient descent* secara tidak sengaja meningkatkan kesalahan daripada mengurangi kesalahan pada saat pelatihan. Oleh karena itu, dapat disimpulkan bahwa semakin besar nilai *learning rate*, maka akan menyebabkan kesalahan dalam melakukan pembaruan bobot yang nantinya akan mempengaruhi hasil akurasi pelatihan.

### C. Pengujian Iterasi Epoch

Percobaan dilakukan dengan *learning rate* 0.001, Model optimasi Adam yang digunakan, *decay* 0.8 dan *dropout* 0.8 [15]. Hal ini dilakukan untuk mengetahui pengaruh ukuran iterasi pada *epoch* terhadap proses pembelajaran dan juga pengujian data. Hasil yang diperoleh dari percobaan *learning rate* dapat dilihat pada Tabel 6.

TABEL 6 PENGUJIAN NILAI EPOCH

No.	Epoch	Data Latih		Data Uji	
		Loss	Accuracy (%)	Loss	Accuracy (%)
1.	50	0.2344	87.22	2.8223	32.63
2.	100	0,0767	92.09	0.6723	56.32
3.	150	0.0748	93.41	0.7617	57.37
4.	200	0.0563	94.16	0.3871	55.26

Pada Tabel 6 menunjukkan perbedaan hasil akurasi yang diperoleh berdasarkan nilai *epoch*. *Epoch* dengan nilai terbesar pada nilai *epoch* 150 dengan hasil nilai akurasi pada data uji sebesar 57.37%, *epoch* dengan nilai 100 sampai 200 tingkat akurasi menjadi lebih baik dibandingkan dengan nilai *epoch* 50 memiliki hasil akurasi yang signifikan.

### KESIMPULAN

Penelitian ini telah melakukan prediksi harga saham menggunakan Recurrent Neural Networks dengan model LSTM. Sistem prediksi harga saham terdiri dari tiga tahap. Tahap pertama yaitu *pra-proses* data yang terdiri dari ekstraksi fitur, normalisasi, *ascending* dan segmentasi. Tahap kedua yaitu proses pelatihan data menggunakan RNN, kemudian

tahap ketiga yaitu proses pengujian. Hasil dari penelitian ini menunjukkan bahwa prediksi harga saham menggunakan RNN mendapatkan hasil kurang baik. Menggunakan optimasi Adam dengan nilai *learning rate* 0.001 dan *epoch* 200 memperoleh akurasi untuk data latih sebesar 94.16% dan data uji sebesar 55.26%. Dapat disimpulkan bahwa jumlah *epoch*, model optimasi dan ukuran *learning rate* untuk proses pelatihan dapat mempengaruhi hasil akurasi yang diperoleh. Oleh karena itu beberapa sarana untuk pengembangan sistem prediksi ini di antaranya menambah faktor-faktor yang berpengaruh terhadap pergerakan harga saham sebagai variasi data latih dan memadukan antara analisis teknikal, fundamental sebagai atribut-atribut pelatihan dan pengujian.

### REFERENSI

- [1] D. Yendriani, M. Si, and U. N. Wisesty, "Prediksi Harga Saham Menggunakan Hidden Markov Model ( HMM ) dan Fuzzy Model," vol. 2, no. 2, pp. 6592–6599, 2014.
- [2] E. Mala, S. Rochman, and A. Djunaidy, "Prediksi harga saham yang mempertimbangkan faktor eksternal menggunakan jaringan saraf tiruan," vol. 1, no. 2, pp. 5–11, 2014.
- [3] L. Troiano, E. M. Villa, and V. Loia, "Replicating a Trading Strategy by Means of LSTM for Financial Industry Applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3226–3234, 2018.
- [4] J. Huang, Y. Zhang, J. Zhang, and X. Zhang, "A Tensor-Based Sub-Mode Coordinate Algorithm for Stock Prediction," 2016.
- [5] J. Poulos, "Predicting Stock Market Movement with Deep RNNs," pp. 1–7, 2013.
- [6] J. B. Heaton, "Deep Learning in Finance," no. February, pp. 1–20, 2016.
- [7] A. Rahmi, W. F. Mahmudy, and B. D. Setiawan, "Prediksi Harga Saham Berdasarkan Data Historis Menggunakan Model Regresi Yang Dibangun Dengan Algoritma Genetika," *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, vol. 5, no. 12, pp. 1–9, 2015.
- [8] M. Velay and F. Daniel, "Stock Chart Pattern recognition with Deep Learning," 2018.
- [9] L. Yu, J. Chen, G. Ding, Y. Tu, J. Yang, and J. Sun, "Spectrum prediction based on taguchi method in deep learning with long short-term memory," *IEEE Access*, vol. 6, no. c, pp. 15923–15933, 2018.
- [10] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video Captioning with Attention-Based LSTM and Semantic Consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, 2017.
- [11] A. Pulver and S. Lyu, "LSTM with working memory," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017–May, pp. 845–851, 2017.
- [12] A. Calvez and D. Cliff, "Deep Learning can Replicate Adaptive Traders in a Limit-Order-Book Financial Market," 2018.
- [13] N. S. Keskar and R. Socher, "Improving Generalization Performance by Switching from Adam to SGD," no. 1, 2017.
- [14] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1–15, 2014.
- [15] A. Greenwald and K. Hall, "Correlated-Q Learning," no. 3, pp. 84–89, 2002.

