

ESTIMASI HARGA KEBUTUHAN POKOK DI KOTA BANDUNG DAN PROVINSI JAWA BARAT MENGGUNAKAN METODE LSTM

BASIC NECESSITIES PRICE ESTIMATION IN BANDUNG CITY AND JAWA BARAT PROVINCE USING LSTM METHOD

Muhammad Refian Sudjarpadi Putra¹, Andrew Brian Osmond², Anton Siswo Raharjo Ansori³

^{1,2,3}Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

¹refianmuhammad@student.telkomuniversity.ac.id, ²abosmond@telkomuniversity.ac.id,

³raharjo@telkomuniversity.ac.id

Abstrak

Fenomena fluktuasi harga khususnya pada harga bahan pokok merupakan pilar utama dari kegiatan ekonomi, yang tentunya akan sangat menguntungkan semua pihak yang terlibat jika terdapat metode untuk memprediksi fluktuasi harga di hari yang akan datang. Maka, dirancang sebuah model pembelajaran mesin untuk memprediksi harga bahan pokok dengan menggunakan metode Long Short-Term Memory Networks (LSTM). Diharapkan dengan metode ini dapat menjadi sebuah sistem yang dapat diandalkan dalam memprediksi harga bahan pokok, baik dari segi akurasi maupun dari segi rentang waktu prediksi di hari yang akan datang.

Kata kunci : Prediksi, Prakiraan, Bahan Pokok, LSTM, Deret Waktu

Abstract

Price fluctuation, especially on basic necessities prices, is a main pillar of economy activities. Which in a sense, it would be very advantageous to every party involved if there is a method to predict the fluctuation of future prices. Therefore, a machine learning model will be designed using Long Short-Term Memory Networks (LSTM) method to predict basic necessity prices. In a hope that this method can be a reliable method, from accuracy wise to prediction time span wise, to predict future values of basic necessities.

Keywords: Prediction, Forecast, Commodity Prices, LSTM, Time Series

1. Pendahuluan

Masyarakat biasa menggunakan intuisi dan berbagai teknik penalaran lainnya untuk memperkirakan harga bahan pokok. Ini membutuhkan pengetahuan yang mendalam tentang harga pasar untuk bisa memperkirakannya secara akurat. Perkiraan harga bahan pokok memberikan pengaruh yang signifikan di bidang ekonomi, politik dan industri [1]. Tetapi harga bahan pokok mempunyai sifat yang sangat fluktuatif dan sulit diperkirakan. Ini berarti tidak ada pola yang konsisten untuk bisa memperkirakan harga pasar dari waktu ke waktu dengan sempurna. Disinilah dimana analisis deret waktu dibutuhkan. Kita membutuhkan model pembelajaran mesin yang bisa melihat data riwayat fluktuasi harga bahan pokok beberapa tahun ke belakang dan memprediksikan harga bahan pokok di waktu yang akan datang dengan tepat.

Pembelajaran mesin atau yang juga dikenal sebagai *Machine Learning* merupakan cabang dari kecerdasan buatan. Pembelajaran mesin adalah studi perancangan dan pengembangan algoritma yang digunakan untuk melakukan perintah tertentu tanpa menggunakan instruksi yang eksplisit, melainkan dengan mengandalkan pola dan inferensi [2]. Algoritma pembelajaran mesin membangun dan mengembangkan model statistik dari data sampel untuk membuat perkiraan dan keputusan [3].

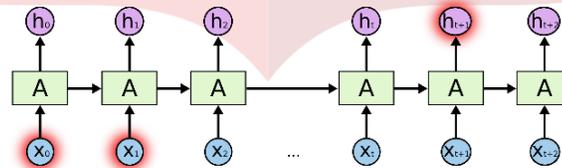
Dalam penelitian ini dibuat sistem yang mampu memperkirakan harga bahan pokok di Kota Bandung dengan memanfaatkan data yang didapat dari Dinas Perdagangan dan Perindustrian Kota Bandung. Sistem ini dibangun berdasarkan metode *Long Short-Term Memory* (LSTM) yang ditujukan

untuk mempelajari data historis dari harga bahan pokok di Kota Bandung beberapa tahun ke belakang. Dengan menggunakan metode ini, diharapkan bisa dibuat sebuah model pembelajaran mesin dengan performa yang cepat dan akurat dalam memperkirakan harga bahan pokok.

2. Dasar Teori

2.1. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan sebuah turunan dari arsitektur *Recurrent Neural Network* (RNN) [4]. Banyak peneliti yang mengembangkan metode LSTM di berbagai bidang seperti dalam bidang prediksi deret waktu atau *forecasting*. RNN tidak mampu lagi untuk belajar menghubungkan informasi ketika ada kesenjangan tumbuh diantara informasi yang relevan. Ketika kesenjangan itu tumbuh, memori yang tersimpan akan semakin tidak relevan seiring waktu berjalan karena tertimpa dengan memori baru, kekurangan ini dipresentasikan oleh Yoshua Bengio (1994) [5, 6]. Pada Gambar 1 [7] diilustrasikan kekurangan RNN yang dapat dilihat dari masukan X_0, X_1 memiliki kesenjangan yang sangat jauh dengan X_{t+1} sehingga ketika H_{t+1} memerlukan informasi yang relevan dengan X_1 , RNN tidak dapat menghubungkan informasi karena kesenjangan yang terlalu jauh.



Gambar 1. Diagram Memori RNN

Di sisi lain, LSTM mampu mengatasi kekurangan tersebut karena metode ini dapat mengatur memori pada setiap masukannya dengan menggunakan *memory cells* dan *gate units* pada setiap *neurons* yang berfungsi sebagai pengatur memori. Terdapat empat fungsi aktivasi disetiap masukan *neurons* yang disebut *gates units*. *Gates units* ini terdiri atas *forget gates*, *input gates*, *cell gates*, dan *output gates*. Pada *forget gates*, setiap masukan akan dipilih mana saja yang akan disimpan atau dibuang. Fungsi aktivasi yang digunakan adalah fungsi sigmoid yang hasil keluarannya antara nilai 0 dan 1. Semua data yang keluarannya bernilai 1 maka disimpan, dan sebaliknya untuk keluaran yang bernilai 0 akan dibuang. Pada *Input gates* terdapat dua proses dimana pertama akan diputuskan nilai yang akan diperbarui menggunakan fungsi sigmoid yang selanjutnya akan membuat vektor nilai baru dan disimpan di *memory cells* menggunakan fungsi tanh. Pada *Cell gates*, nilai lama yang ada di *memory cells* akan diganti dengan nilai *memory cells* yang baru. Nilai yang baru didapatkan dari gabungan antara nilai yang terdapat di *forget gate* dan *input gate*. Terakhir, Pada *output gates* akan diputuskan nilai pada bagian *memory cell* mana yang akan dikeluarkan dengan menggunakan fungsi sigmoid yang selanjutnya nilai akan ditempatkan pada *memory cell* dengan menggunakan fungsi tanh dan kedua *gates* tersebut dikalikan sehingga menghasilkan nilai keluaran. Berikut adalah rumus matematis dari *LSTM*:

$$\begin{aligned}
 i_t &= \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \\
 f_t &= \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \\
 \check{C}_t &= \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \\
 C_t &= f_t \cdot C_{t-1} + i_t \cdot \check{C}_t \\
 h_t &= b_t \cdot \tanh(C_t) \\
 o_t &= \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)
 \end{aligned}
 \tag{1}$$

Dimana :

σ : Fungsi sigmoid.

i_t : Input gate.

f_t : Forget gate.

x_t : Input pada current timestamp.

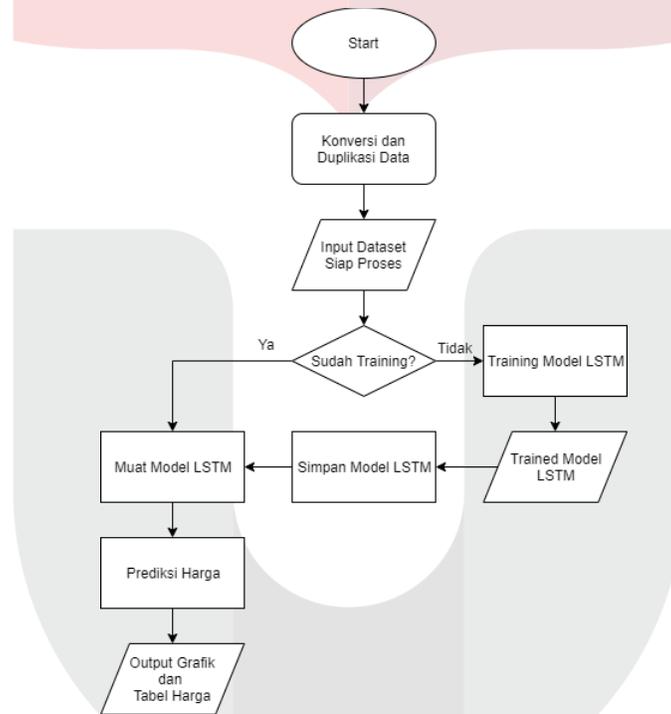
w_x : Bobot neuron pada masing-masing gate.

h_{t-1} : Output dari pada (t-1)
 b_x : Bias pada masing-masing gate.
 C_t : Memory cell pada (t)
 \check{C}_t : Kandidat Memory cell pada (t)
 o_t : Output gate

3. Analisis dan Perancangan Sistem

3.1. Perancangan Sistem Keseluruhan

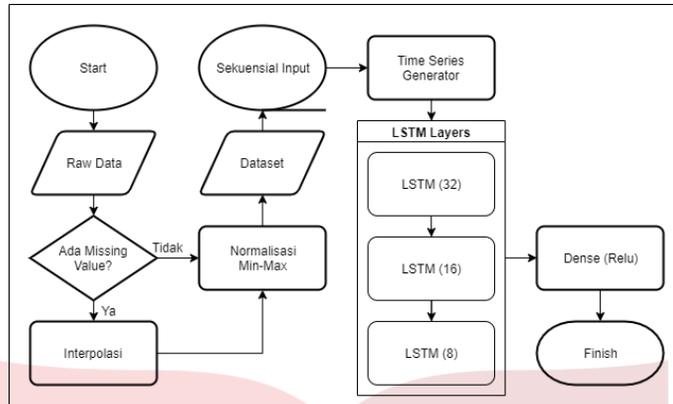
Memperoleh dataset yang mentah dari Dinas Perindustrian dan Perdagangan Provinsi Jawa Barat (DISPERINDAG). Lalu dilakukan proses konversi dan duplikasi data secara manual dengan keluaran merupakan dataset siap proses. Lalu program memulai *Training Model LSTM* untuk mempelajari pola pergerakan Harga Bahan Pokok. Keluaran dari tahap ini merupakan *Trained Model LSTM*. Serta menyimpan *Trained Model LSTM* dalam format HDF5. Program melakukan prediksi harga dan keluaran terakhir merupakan Grafik dari harga bahan pokok yang telah diprediksi.



Gambar 2. Diagram Alur Sistem Secara Keseluruhan

3.2. Perancangan Model LSTM

Program dirancang dengan bahasa pemrograman python 3 dan menggunakan *neural networks* API yang bernama Keras. Masukan program merupakan data harga historis dari tahun 2014 sampai pertengahan tahun 2019 dengan total panjang dataset sebesar 2061 baris data untuk setiap 33 komoditas yang tersedia. Data di-normalisasi menggunakan metode MinMax. Hal ini dilakukan untuk menjaga kestabilan proses training. Lalu dataset dirubah dari bentuk sekuensial ke bentuk *supervised time series*. Datsset lalu di-*feed* ke layer pertama LSTM dengan 32 buah neurons, ke layer kedua dengan 16 buah neurons, dan ke layer terakhir dengan 8 buah neurons. Lalu model di-*training* menggunakan *Adamax Optimizer* dengan perhitungan loss menggunakan metode *Mean Squared Error*.



Gambar 3. Diagram Model LSTM

4. Pengujian Sistem

4.1. Pengujian Model

Ada 4 parameter yang akan diuji untuk setiap komoditas, yang terdiri dari Nilai Epoch, Nilai Lookback, Jumlah Hidden Layer, dan Nilai Train Batch. Nilai Epoch adalah banyaknya perulangan pada tahap training dimana di setiap iterasi perulangan akan menghasilkan generasi neural network yang lebih baik dari iterasi sebelumnya. Nilai Lookback adalah banyaknya rentang hari yang dilihat kebelakang sebagai pola untuk menentukan nilai pada H_0 . Nilai Train Batch adalah banyaknya data yang dibaca dalam satu runtutan data pada setiap iterasi. Jumlah hidden layer adalah banyaknya layer LSTM pada model, terdapat *Single-Layer* dan *Multi-Layer* dimana *Multi-Layer* sendiri terdiri dari 2 dan 3 hidden layer. Pengujian nilai epoch dan nilai lookback akan dilakukan dalam satu kelompok pengujian yang berarti tahap pengujian akan dibedakan berdasarkan jumlah hidden layer pada masing masing nilai Train Batch. Pada Tabel 1 adalah 6 sampel konfigurasi terbaik dari total 290 konfigurasi yang diuji pada komoditas Daging Ayam Broiler. Nilai loss pada tabel dihitung dengan menggunakan metode *mean squared error*.

Tabel 1. Hasil Pengujian Pada Komoditas

No	Train Batch	LSTM Hidden Layers	Lookback	Epoch	Loss
1	3	3	14	128	0.001154
2	7	3	28	256	0.001155
3	7	3	7	256	0.001165
4	1	3	28	128	0.001178
5	1	3	28	64	0.001182
6	3	2	21	128	0.001188

4.2. Pengujian Prediksi

Tujuan pengujian prediksi disini yaitu untuk menguji tingkat akurasi dan kecocokan model LSTM dalam memprediksi harga masa depan dengan menggunakan konfigurasi dari hasil pada pengujian model. Pengujian prediksi dengan bobot konfigurasi Train Batch yang bernilai 3, dengan 3 Hidden Layer LSTM, 14 Hari Lookback, dan 128 Epoch. Berikut pada gambar 4 adalah grafik hasil prediksi selama 30 hari.



Gambar 4 Hasil Prediksi Komoditas Daging Ayam Broiler

Dari hasil pada gambar 4, dapat dilihat bahwa hasil prediksi masih jauh dari yang diharapkan, meskipun model mampu menghasilkan nilai yang mendekati akurat pada tahap training. Namun jika diperintah untuk memprediksi nilai masa depan, hal ini mungkin terjadi karena pergerakan harga pada komoditas-komoditas bahan pokok di Jawa Barat mempunyai pola yang sangat mengacu pada hari lebaran yang berarti pola hanya terulang setiap tahun dimana harga tidak selalu berulang dengan pola yang sama persis pada tahun-tahun sebelumnya.

5. Kesimpulan

Ada empat parameter yang bisa di-tuning agar proses training dan proses prediksi yang didapat seakurat mungkin, yaitu nilai epochs, nilai lookback, nilai train batch dan jumlah layer LSTM. Banyaknya nilai epoch yang diberikan proporsional dengan akurasi yang dihasilkan, dimana semakin besar nilai epoch yang diberikan maka semakin akurat hasil yang dihasilkan pada saat proses training model, dengan waktu training yang dibutuhkan juga ikut semakin besar. Banyaknya nilai train batch yang diberikan akan berbanding lurus dengan akurasi yang dihasilkan, dimana semakin kecil maka semakin akurat hasil yang dihasilkan, dengan waktu training yang dibutuhkan juga semakin membesar. Banyaknya jumlah layer LSTM dan nilai lookback yang diberikan tidak proporsional dengan tingkat akurasi yang dihasilkan, yang membuat kedua parameter ini menjadi redundant dan membutuhkan manual tuning. Dataset yang digunakan tidak terlalu menunjukkan nilai trend musiman atau *seasonal trend*, yang berarti data yang digunakan memiliki data yang tidak terduga, sangat kontras dengan kemampuan dari metode LSTM. Maka disimpulkan bahwa LSTM sudah mampu mempelajari pergerakan data, menggunakan harga bahan pokok sebagai subjek data, namun dalam implementasi prediksi masa depan, LSTM masih jauh dari harapan.

Daftar Pustaka:

- [1] V. E. Satya, "Kajian Singkat Terhadap Anomali Fluktuasi Harga Bahan Pangan di Indonesia," Pusat Penelitian Badan Keahlian, Jakarta, 2016.
- [2] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, London: MIT Press, 2012.
- [3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Ann Arbor: University of Michigan, 1994.
- [4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [5] Y. Bengio, "The Problem of Learning Long-Term Dependencies in Recurrent Networks," in *Neural Networks for Computing Conference*, Snowbird, 1993.
- [6] Y. Bengio, P. Frasconi and P. Simard, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157-166, 1994.
- [7] C. Olah, "Understanding LSTM Networks," 27 August 2015. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed 28 February 2019].