

Analisis dan implementasi Sistem Trading Otomatis Bitcoin dengan Menggunakan Recurrent Reinforcement Learning

Bintang Aryo Dharmawan¹, Jondri²

^{1,2}Fakultas Informatika, Universitas Telkom, Bandung

¹bintanglekstrim@student.telkomuniversity.ac.id, ²jondri@telkomuniversity.ac.id

Abstrak

Bitcoin adalah salah satu *cryptocurrency* yang diminati untuk menjadi media investasi dalam mencapai keuntungan finansial. Meskipun investasi menggunakan Bitcoin sangat populer, investasi jenis ini memiliki risiko yang perlu dipertimbangkan. Untuk mengantisipasi risiko dalam berinvestasi menggunakan Bitcoin, sistem perdagangan diperlukan untuk berdagang secara otomatis. Sistem dibangun menggunakan dua metode komputasi, yaitu *Recurrent Neural Network* dan metode *Reinforcement Learning* yang kemudian disebut *Recurrent Reinforcement Learning*. Metode ini memerlukan nilai parameter yang tepat untuk memaksimalkan nilai *sharpe ratio*. Nilai *sharpe ratio* digunakan untuk mengukur kelebihan pengembalian, atau premi risiko, per unit deviasi dalam aset investasi atau strategi perdagangan. Dalam penelitian tugas akhir ini, dilakukan analisis terhadap parameter yang mempengaruhi kinerja sistem. Hasil yang diperoleh dari analisis yang telah dilakukan menyatakan bahwa sistem mendapatkan nilai *sharpe ratio* 0,10963. Nilai *sharpe ratio* yang didapatkan masih relatif tinggi karena suatu investasi dinilai beresiko rendah jika nilai *sharpe ratio* nya satu keatas.

Kata Kunci:*Bitcoin, Trading, Recurrent Reinforcement Learning, RRL*

Abstract

Bitcoin is one of the *cryptocurrency* in demand to be an investment medium in achieving financial returns. Although investing using Bitcoin is very popular, this type of investment has risks that need to be considered. To anticipate risks in investing using Bitcoin, a trading system is needed to trade automatically. The system is built using two computational methods, namely *Recurrent Neural Network* and *Reinforcement Learning* method which is then called *Recurrent Reinforcement Learning*. This method requires the right parameter values to maximize the *sharpe ratio* value. *Sharpe ratio* values are used to measure excess returns, or risk premiums, per unit deviation in investment assets or trading strategies. In this final project research, an analysis of parameters that affect system performance is carried out. The results obtained from the analysis that has been done states that the system gets a *sharpe ratio* value of 0,10963. The value of the *sharpe ratio* obtained is still relatively high because an investment is considered low risk if the value of the *sharpe ratio* is one and above.

Keywords: *Bitcoin, Trading, Recurrent Reinforcement Learning, RRL*

1. Pendahuluan

Latar Belakang

Salah satu pendekatan yang relatif baru dalam perdagangan keuangan adalah menggunakan algoritma pembelajaran mesin untuk memprediksi kenaikan dan penurunan harga aset sebelum terjadi. Trader yang optimal akan membeli aset sebelum harga naik, dan menjual aset sebelum nilainya menurun. *Recurrent Reinforcement Learning* (RRL) menyesuaikan parameter suatu sistem untuk memaksimalkan hasil atau hadiah yang diharapkan yang dihasilkan karena tindakan dari sistem. Sistem RRL menggunakan agent yang berfungsi untuk menentukan tindakan atau aksi yang akan dilakukannya. Disini agent akan selalu memberikan tindakan atau aksi yang bisa disebut jual atau beli lembar saham sampai jangka waktu yang ditentukan oleh penulis. Ketika sistem RRL telah mendapatkan *reward* atau hasil yaitu profit yang diinginkan maka penulis mendapatkan parameter untuk sistem RRL tersebut yang nantinya akan selalu dipakai sampai mencapai profit yang maksimal. Dalam tugas akhir ini penulis akan melakukan penelitian mengenai Analisis harga Bitcoin untuk ditransaksikan menggunakan metode RRL secara otomatis. Dengan menggunakan metode tersebut diharapkan dapat menghasilkan profit yang maksimal.

Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, didapatkan rumusan masalah yang diangkat pada tugas akhir ini, yaitu Bagaimana mengimplementasikan RRL untuk trading Bitcoin dan Bagaimana performansi arsitektur RRL dalam trading Bitcoin.

Tujuan

Tujuan dari Tugas Akhir ini adalah Menganalisis dan mengimplementasikan arsitektur RRL dalam trading Bitcoin dan Menganalisis akurasi sistem yang menggunakan arsitektur RRL dalam trading Bitcoin.

Organisasi Tulisan

Tugas Akhir ini disusun dengan struktur sebagai berikut. Setelah dijelaskan pendahuluan pada bagian pertama, dijelaskan studi terkait pada bagian kedua. Selanjutnya, dijelaskan pemodelan sistem pada bagian ketiga. Setelah itu, evaluasi performansi sistem terhadap sistem yang dibangun pada bagian ketiga. Bagian terakhir, dijelaskan kesimpulan dan saran untuk penelitian selanjutnya pada bagian keempat.

2. Studi Terkait

2.1 Penelitian Terkait

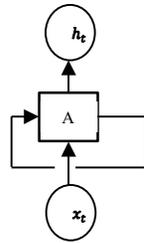
Terdapat beberapa karya tulis yang terkait dengan tulisan ini seperti “*Stock Trading with Recurrent Reinforcement Learning*” dan “*FX Trading with Recurrent Reinforcement Learning*” [1]. Karya tulis tersebut membahas tentang menganalisis harga stock market dengan sistem RRL dan *Stock Tading* pada *Stock Market*. Pada Karya yang berjudul “*FX Trading with Recurrent Reinforcement Learning*” [1], Penelitian tersebut juga memberi penjelasan tentang sistem trading yang menggunakan metode *Recurrent Neural Network*. Dalam karya tulis tersebut, menggunakan *sharpe ratio* untuk mengevaluasi hasil performa latihan dari sistem trading otomatis ini sebagai pembanding input dan hasil dari algoritma RRL.

2.2 Recurrent Neural Network

Manusia tidak memulai pemikiran mereka dari goresan awal di setiap detiknya. Ketika manusia membaca, pemahaman setiap katanya terjadi berdasarkan pemahaman dari kata – kata sebelumnya. Kata – kata sebelumnya tidak akan diabaikan begitu saja dan mulai pemikiran dari awal lagi. Pemikiran manusia memiliki ketekunan untuk melakukan ini. [2]

Traditional neural networks tidak bisa melakukan hal ini, dan ini menjadi kelemahan utama. Sebagai contoh, ketika kita memiliki suatu seri data. Dengan menggunakan *traditional neural network* tidak dapat digambarkan bagaimana sebuah seri data sebelumnya dapat menggambarkan seri data berikutnya.

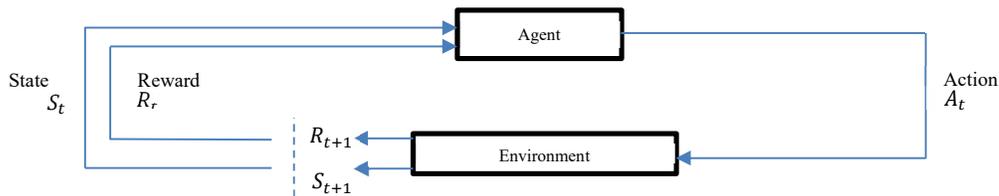
Recurrent neural network hadir untuk menyelesaikan permasalahan ini, dengan adanya hubungan yang berulang didalamnya, sehingga memungkinkan informasi untuk bertahan. [2]



Gambar 2.1 arsitektur RNN [2]

2.3 Reinforcement Learning

Reinforcement learning merupakan bagian dari salah satu teknik *machine learning* dimana terdapat sebuah agent yang melakukan sebuah aksi dan belajar dari hasil aksi tersebut (belajar berdasarkan pengalaman yang terjadi pada agent tersebut).



Gambar 2.2 arsitektur Reinforcement Learning [4]

Berdasarkan pada gambar 2.2 Menjelaskan bagaimana alur kerja metode *reinforcement learning*. Proses belajar dari metode *reinforcement learning* adalah pertama *Agent* menerima sebuah *State* (S_t) dari *Environment* yang ada, selanjutnya agent akan melakukan sebuah *Action* (A_t). *Environment* akan berubah menjadi *State* baru (S_{t+1}) dan memberikan sebuah *Reward* (R_{t+1}) pada agent. Proses tersebut akan dilakukan berulang, hingga *agent* akan belajar untuk memaksimalkan kumulatif reward yang didapat. [3]

2.4 Recurrent Reinforcement Learning

Tujuan dari *Recurrent Reinforcement Learning* adalah untuk memperbaharui nilai pembobotan fungsi aktivasi yang berada pada agent sebagai trader dengan menggunakan *gradient ascent* seperti persamaan di bawah ini.

$$w_t = w_{t-1} + r \frac{dS_t}{dw_t} = w_{t-1} + \Delta w \tag{2.1}$$

Dimana w_t sebagai bobot dalam jaringan pada waktu t , r adalah nilai *learning rate* yang disesuaikan, dan S_t adalah perhitungan performansi dalam sebuah sistem trading. Dalam penelitian ini digunakan perhitungan *sharpe ratio* untuk mengukur performansi sistem trading.

2.5 Sharpe Ratio

Sharpe ratio adalah ukuran yang digunakan untuk mengukur kelebihan pengembalian, atau premi risiko, per unit deviasi dalam aset investasi atau strategi perdagangan. Pada penelitian ini *sharpe ratio* digunakan sebagai alat ukur untuk mengukur proses pengelolaan investasi. Jika *Sharpe ratio* digunakan dengan benar, dapat meningkatkan proses pengelolaan investasi. [4]

$$S_T = \frac{\text{Average}(R_t)}{\text{Standard Deviation}(R_t)} \text{ for interval } t = 1, \dots, T \tag{2.2}$$

Dimana R_t adalah hasil *return* dari investasi dalam periode trading t . *Sharpe ratio* ini digunakan sebagai *reward* pada setiap langkah investasi yang akan dibuat. Semakin tinggi nilai *sharpe ratio* maka semakin tinggi nilai *reward* yang didapatkan, begitupun sebaliknya. [1]

2.6 Fungsi Trading

Trader akan berusaha untuk memaksimalkan nilai *sharpe ratio* pada setiap langkahnya. Pada penelitian ini, fungsi trader yang digunakan untuk pengambilan keputusan adalah metode *recurrent neural network* di mana tanh adalah fungsi aktivasi dalam setiap neuron jaringannya [1]:

$$F_t = \tanh(w^T x) \quad (2.3)$$

Inputan yang diberikan adalah $x_t = [1, r_t, \dots, r_{t-M}, F_{t-1}]$, M adalah jumlah nilai data time series yang diinputkan ke trader dan hasil *return* $r_t = p_t - p_{t-1}$. Di mana hasil *return* (r_t) didapatkan dari periode (t) dikurang dengan periode sebelumnya ($t-1$). Dengan nilai keluarannya adalah $[-1, 1]$, di mana jika $F_t < 0$ maka akan melakukan tindakan menjual, jika $F_t > 0$ akan membeli, dan jika $F_t = 0$ tidak menjual atau membeli. Di mana μ adalah banyaknya saham pada tiap transaksi. Pada pengambilan waktu t , pengambilan keputusan F_{t-1} dijelaskan pada persamaan berikut :

$$R_t = \mu \cdot (F_{t-1} \cdot r_t - d |F_t - F_{t-1}|) \quad (2.4)$$

Di mana d adalah biaya transaksi pada setiap periode t . Jika $F_t = F_{t-1}$ (tidak ada perubahan nilai investasi pada periode ini) maka tidak ada biaya tambahan transaksi. Sisi lain nilai biaya tambahan sebanding dengan selisih kepemilikan saham.

2.7 Gradient Ascent

Untuk memaksimalkan nilai *sharpe ratio* diperlukan gradient ascent. Pertama definisikan fungsi utilitas menggunakan rumus dasar statistika untuk rata - rata dan variansi dengan persamaan di bawah ini :

$$S_T = \frac{E[R_t]}{\sqrt{E[R_t^2] - (E[R_t])^2}} = \frac{A}{\sqrt{B - A^2}} \quad \text{dimana } A = \frac{1}{T} \sum_{t=1}^T R_t \quad \text{dan } B = \frac{1}{T} \sum_{t=1}^T R_t^2 \quad (2.5)$$

Kemudian fungsi di atas dapat dilakukan penurunan dari S_T menggunakan fungsi sebagai berikut :

$$\begin{aligned} \frac{dS_T}{dw} &= \frac{d}{dw} \left\{ \frac{A}{\sqrt{B - A^2}} \right\} = \frac{dS_T}{dA} \cdot \frac{dA}{dw} + \frac{dS_T}{dB} \cdot \frac{dB}{dw} \\ &= \sum_{t=1}^T \left\{ \frac{dS_T}{dA} \cdot \frac{dA}{dR_t} + \frac{dS_T}{dB} \cdot \frac{dB}{dR_t} \right\} \cdot \frac{dR_t}{dw} = \sum_{t=1}^T \left\{ \frac{dS_T}{dA} \cdot \frac{dA}{dR_t} + \frac{dS_T}{dB} \cdot \frac{dB}{dR_t} \right\} \cdot \left\{ \frac{dR_t}{dF_t} \cdot \frac{dF_t}{dw} + \frac{dR_t}{dF_{t-1}} \cdot \frac{dF_{t-1}}{dw} \right\} \end{aligned} \quad (2.6)$$

3. Sistem yang Dibangun

Pada bagian ini, akan dijelaskan secara jelas tentang metode Recurrent Reinforcement Learning yang diusulkan. Langkah-langkah membangun sistem yang dibuat terdiri dari inialisasi parameter, training RRL, dan melakukan uji terhadap data testing. Dilakukan inialisasi pada setiap parameter, setelah itu dilakukan training pada RRL yang dibuat sesuai dengan parameter yang telah ditentukan. Selanjutnya dilakukan uji pada model yang telah didapatkan dari proses training terhadap data testing. Proses tersebut terus diulang hingga mendapatkan model dengan akurasi yang dapat diterima.

3.1 Inialisasi Parameter

Ditentukan inialisasi parameter-parameter dasar yang dibutuhkan yang akan dianalisis antara lain :

- Nilai *Learning rate*
- Epoch* Maksimum
- jumlah nilai data time series yang diinputkan ke trader
- dataset pada setiap kondisi market yang berbeda

3.3 Training RRL

Metode dengan tingkat kesuksesan yang tinggi menggunakan algoritma sebagai berikut :

1. Lakukan training parameter $w \in R^{M+2}$ menggunakan historis data pada rentan waktu T
2. Gunakan nilai w yang paling optimal untuk membuat pengambilan keputusan dalam kondisi waktu yang sebenarnya dari $t = T + 1$ hingga $t = T + N_{prediksi}$
3. Setelah prediksi $N_{prediksi}$ lengkap, ulangi langkah pertama.

3.4 Testing

Model yang telah didapatkan pada proses training akan diuji dengan menggunakan data testing yang telah didapat dari *preprocessing data*, dengan acuan *Sharpe ratio* yang didapat dari hasil pengujian pada data testing.

4. Evaluasi

4.1 Dataset

Pada penelitian ini dataset yang digunakan menggunakan data jual beli indeks harga Bitcoin terhadap IDR yang diambil dari luno.com website (dapat diakses pada https://www.cryptodatadownload.com/cdd/Luno_BTCUSD_d.csv), data yang digunakan pada tanggal 17/12/2017 – 19/05/2018, 17/09/2017 - 17/12/2017, 4/12/2018 - 4/03/2019. Setelah itu, dibagi menjadi data training dan data testing dengan komposisi 70:30.

4.2 Skenario Pengujian

Skenario pengujian dalam Tugas Akhir ini adalah menganalisa dampak parameter terhadap nilai Sharpe ratio yang didapatkan. Parameter yang diuji berupa dataset pada setiap kondisi market yang berbeda, nilai *Learning rate*, besarnya max epoch, jumlah nilai data time series yang diinputkan ke trader untuk menghasilkan nilai bobot yang optimal. Berikut adalah nilai parameter yang akan diuji :

1. Jumlah nilai data time series yang diinputkan ke model : 7, 14, 21, 30, 90
2. Max Epoch : 100, 500, 1000, 5000, 5500, 10000
3. Nilai *learning rate*: 0.1, 0.25, 0.5, 0.75, 1.0
4. Dataset pada setiap kondisi : market turun (1/11/2017 – 15/12/2018), kondisi market naik (17/09/2017 - 17/12/2017), kondisi market stagnan (4/12/2018 - 4/03/2019)

Untuk setiap parameter dikombinasikan dan dilakukan observasi sebanyak 5 kali. Hal ini dilakukan karena bobot awal dihasilkan secara *random*, sehingga tidak cukup dilakukan observasi hanya sekali saja, karena hasil yang didapatkan bisa saja kebetulan baik ataupun buruk.

4.3 Menganalisis parameter jumlah nilai data time series yang diinputkan

Parameter jumlah nilai data time series yang diinputkan pada model dalam tugas akhir ini adalah 1, 2, 3, 4, 5, 6, 7, 14, 30

Tabel 4.1 Kombinasi Parameter Nilai Data Input

Nilai Data Input	Learning Rate	Epoch	Sharpe Ratio
1	0.1	1000	0.09817
2	0.1	1000	0.07861
3	0.1	1000	0.09410
4	0.1	1000	0.11111
5	0.1	1000	0.10249
6	0.1	1000	0.91824
7	0.1	1000	0.10772
8	0.1	1000	0.86405
14	0.1	1000	0.09053
30	0.1	1000	0.07200

Berdasarkan Tabel 4.1, menunjukkan bahwa jumlah nilai data input 4 menghasilkan hasil yang lebih optimal dibandingkan dengan nilai data input lainnya. Walaupun jumlah nilai data input 4 dan nilai data input 7 memiliki selisih 0.00339 tetapi dalam *sharpe ratio* itu memiliki nilai yang besar karena 0.0039 dikalikan dengan harga Bitcoin per US Dolarnya. Jumlah nilai data input merepresentasikan banyaknya pola yang akan dipelajari RRL. Tidak ada aturan pasti mengenai jumlah banyaknya nilai data input yang paling optimal dalam RRL. Sehingga jumlah nilai data input yang didapatkan melalui percobaan.

4.4 Menganalisis parameter besarnya max epoch

Parameter max epoch yang digunakan pada tugas akhir ini adalah 100,500, 1000, 5500, 10000

Tabel 4.2 Kombinasi Parameter Epoch

Nilai Data Input	Learning Rate	Epoch	Sharpe Ratio
4	0.1	100	0.08932
4	0.1	500	0.10908
4	0.1	1000	0.11111
4	0.1	4500	0.11232
4	0.1	5500	0.11232
4	0.1	6000	0.11233
4	0.1	7500	0.11234
4	0.1	8700	0.11235
4	0.1	9500	0.11235
4	0.1	10400	0.11235

Berdasarkan Tabel 4.3, menunjukkan bahwa max epoch 8700 menghasilkan hasil yang lebih optimal dibandingkan dengan nilai data input lainnya. Terlihat dari epoch lebih dari 8700 keatas *sharpe ratio* mengalami stagnant tidak ada lagi perubahan nilai yang signifikan. Tidak ada aturan pasti mengenai *epoch* yang paling optimal dalam RRL. Sehingga max *epoch* yang didapatkan melalui percobaan.

4.5 Menganalisis parameter nilai *learning rate*

Parameter jumlah nilai *learning rate* yang digunakan pada tugas akhir ini adalah 0.0001, 0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 1.0

Tabel 4.3 Kombinasi Parameter Learning Rate

Nilai Data Input	Learning Rate	Epoch	Sharpe Ratio
4	0.0001	8700	0.08979
4	0.001	8700	0.08980
4	0.01	8700	0.08985
4	0.1	8700	0.11235
4	0.25	8700	0.08932
4	0.5	8700	0.09143
4	0.75	8700	0.08932
4	1.0	8700	0.08932

Berdasarkan Tabel 4.2, menunjukkan bahwa *learning rate* 0.1 menghasilkan hasil yang lebih optimal dibandingkan dengan nilai data input lainnya. Walaupun *learning rate* 0.1 dan *learning rate* 0.5 memiliki selisih 0.02092 tetapi dalam *sharpe ratio* itu memiliki nilai yang besar karena 0.02092 dikalikan dengan harga Bitcoin[5] per US Dolarnya. *Learning rate* merepresentasikan seberapa besar dominan perubahan tiap bobot. Tidak ada aturan pasti mengenai nilai learning yang paling optimal dalam RRL. Sehingga *learning rate* yang didapatkan melalui percobaan.

4.6 Menganalisis parameter dataset pada tiap kondisi market

Parameter dataset pada kondisi market yang digunakan pada tugas akhir ini adalah kondisi *bearish*, *bullish*, *stagnant*, dan *mix*

Tabel 4.4 Kombinasi Parameter Kondisi Market

Kondisi Market	Nilai Data Input	Learning Rate	Epoch	Sharpe Ratio
Bearish	4	0.1	8700	0.10963
Bullish	4	0.1	8700	0.04767
Stagnant	4	0.1	8700	0.04820
Mix	4	0.1	8700	0.08766

Berdasarkan Tabel 4.4, menunjukkan bahwa kondisi market *Bearish* menghasilkan hasil yang lebih optimal dibandingkan dengan kondisi market lainnya. Walaupun kondisi market *Bearish* dan *Mix* memiliki selisih 0.02197 tetapi dalam *sharpe ratio* itu memiliki nilai yang besar karena 0.02197 dikalikan dengan harga Bitcoin

per US Dolarnya. Tidak ada aturan pasti mengenai kondisi market apa yang paling optimal dalam metode RRL. Sehingga jumlah nilai data input yang didapatkan melalui percobaan.

5. Kesimpulan

Pada Tugas Akhir ini, dibangun model RRL untuk melakukan trading dengan pengujian parameter nilai data input, *learning rate*, *epoch*, dan pada setiap kondisi market. Pada pengujian tersebut didapatkan hasil yang terbaik yaitu dengan komposisi nilai data input 4, *learning rate* 0.1, *epoch* 8700, dan pada kondisi market *bearish* dengan nilai sharpe ratio 0.10963.

Saran untuk penelitian selanjutnya, yaitu dapat membangun model RRL dengan menambahkan layer atau fitur baru dalam neuron agent dan proses optimasi baru yang digunakan.

6. Daftar Pustaka

- [1] Carl Gold, 2003, "FX Trading via Recurrent Reinforcement Learning", Pasadena, IEEE
- Gabriel Molina, "Stock Trading with Recurrent Reinforcement Learning (RRL)" [PDF Dokumen], diakses : <http://cs229.stanford.edu/proj2006/Molina-StockTradingWithRecurrentReinforcementLearning.pdf>
- [2] Christopher Olah, 2015, "Understanding LSTM Networks", <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>, diakses : Desember, 2017
- [3] Kevi Chandra, Irene Anindaputri, "Reinforcement Learning pada Alphago", <https://socs.binus.ac.id/2018/12/07/reinforcement-learning-pada-alphago/>, diakses : Maret, 2019
- [4] William F.Sharpe, "The Sharpe Ratio", <http://web.stanford.edu/~wfsSharpe/art/sr/sr.htm>, diakses : Januari, 2020
- [5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2009.

Lampiran