DAFTAR TABEL

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Machine learning merupakan salah satu aplikasi kecerdasan buatan *(Artificial Intelligence)* yang digunakan untuk menyelesaikan permasalahan yang memiliki pola, tidak diketahui fungsi pastinya, dan tersedia banyak data [1]. Permasalahan saham memenuhi semua kriteria yang dapat diselesaikan dengan *machine learning*. *Machine learning* dibagi menjadi tiga bagian utama yaitu, *supervised learning, unsupervised learning,* dan *reinforcement learning.*

Reinforcement learning merupakan sub dari *machine learning* yang belajar melalui eksperimen untuk menemukan *action* seperti apa yang akan menghasilkan *reward* paling banyak [2]. Terdapat perluasan dari *reinforcement learning* yang menggabungkan antara deep learning dan *reinforcement learning*, disebut dengan *Deep Reinforcement Learning* (DRL) [3].

Terdapat banyak penelitian yang mengimplementasikan *Deep Reinforcement Learning* di kasus jual-beli saham. Penelitian-penelitian tersebut menyatakan bahwa *Deep Reinforcement Learning* memiliki kinerja yang bagus saat diimplementasikan pada saham India [4], 30 Dow Jones stocks [5], Cina dan S&P 500 [6]. Bahkan penelitian tersebut menyatakan bahwa DRL mampu bekerja dengan baik meskipun diterapkan pada pasar saham yang kompleks, tidak menentu, dan cepat berubah atau dinamis [4][7][8][9].

Sedangkan, beberapa penelitian menunjukkan bahwa terdapat relasi negatif antara *volatility* dengan *expected return* [10][11].

Oleh karena itu, perlu dibuktikan stabilitas kinerja DRL dalam menghadapi volatilitas pasar yang beragam. Sehingga, peneliti tertarik untuk membuktikan stabilitas kinerja DRL dengan mengimplementasikan DRL pada berbagai macam fraksi saham di Indonesia tanpa memasukkan data saham yang terdampak pandemi, dan menerapkan DRL untuk berbagai kasus kombinasi penggunaan data

pandemi pada *data training* dan *data trading.* Dengan demikian, dapat diketahui sejauh mana DRL dapat memberikan kinerja yang baik pada pengimplementasian jual-beli saham pada kondisi pasar yang beragam.

1.2. Rumusan Masalah

Rumusan masalah pada penelitian ini yaitu sebagai berikut:

1. Bagaimana stabilitas algoritma-algoritma *Deep Reinforcement Learning* dalam menangani data saham tiap-tiap fraksi di luar kondisi pandemi?
2. Bagaimana stabilitas algoritma-algoritma *Deep Reinforcement Learning* dalam menangani data saham yang terdampak pandemi COVID-19?

1.3. Tujuan

Tujuan dari penelitian ini dilakukan adalah:

1. Mengetahui seberapa stabil algoritma-algoritma *Deep Reinforcement Learning* dalam menangani data saham tiap-tiap fraksi di luar kondisi pandemi.
2. Mengetahui seberapa stabil algoritma-algoritma model *Deep Reinforcement Learning* dalam menangani data saham yang terdampak pandemi COVID-19.

1.4. Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Pengambilan keputusan dengan *Deep Reinforcement Learning* hanya ditujukan untuk pembelajaran, bukan untuk melakukan perdagangan dengan uang sungguhan.
2. Penelitian ini hanya ditujukan untuk melihat stabilitas algoritma *Deep Reinforcement Learning* terbaik, tidak melakukan *live trading.*
3. Penelitian ini hanya berfokus pada penerapan data saham di Indonesia.
4. Algoritma *Deep Reinforcement Learning* yang digunakan *Proximal Policy Optimization* (PPO), *Deep Deterministic Policy Gradient* (DDPG).

1.5. Metode Penelitian

Metode yang digunakan pada pengerjaan Tugas Akhir ini yaitu:

1. Studi Literatur

   Pencarian data dilakukan pada berbagai sumber jurnal, artikel, buku, dan video pembelajaran. Adapun jurnal utama yang digunakan adalah berbagai jurnal yang menggunakan *library* FinRL. Artikel yang digunakan adalah artikel-artikel yang membahas *Deep Reinforcement Learning* (DRL). Buku yang digunakan adalah berbagai buku perihal saham. Sedangkan, video yang digunakan adalah video pembelajaran yang membahas konseptual dari DRL.

2. Simulasi

   Simulasi dilakukan di atas *environment* OpenAI gym-style, dengan menginisialisasikan sejumlah uang, menggunakan berbagai macam model algoritma DRL.

3. Pengukuran Empirik

   Hasil dari simulasi perdagangan saham (jual/beli saham) atau informasi yang diperoleh dari observasi/penelitian tersebut dicatat. Pencatatat dilakukan pada masing-masing kondisi yang diteliti, dan masing-masing model algoritma DRL yang diteliti.

4. Analisis Statistik

   Dari hasil yang diperoleh, kemudian data diolah seperti dilakukan pemeriksaan, dan pemodelan data untuk dilakukan analisis sehingga dapat diketahui model algoritma DRL mana yang terbaik untuk masing-masing kondisi. Analisis statistik dilakukan setelah pengukuran empirik dilakukan secara berulang-ulang untuk memperoleh angka ketidakpastian hasil pengujian akhir. Semakin banyak pengulangan pengukuran empirik maka semakin kecil angka ketidakpastian hasil pengujian akhir, dan semakin besar nilai hasil pengujian mendekati nilai sebenarnya. Analisis statistik yang digunakan adalah nilai rata-rata.

.

1.6 Ringkasan Sistematika Penulisan

Ringkasan sistematika laporan yaitu sebagai berikut:

Bab I Pendahuluan

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metode penelitian, dan rincian sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini menjelaskan mengenai penelitian terdahulu, teori, dan konseptual hal-hal yang berkaitan dnegan penelitian.

Bab III Perancangan Sistem

Bab ini menjelaskan mengenai perancangan sistem yang dibuat untuk mencapai tujuan dari penelitian.

Bab IV Hasil dan Analisis

Bab ini menjelaskan mengenai hasil pengujian dan analisis pengujian dari penelitian.

Bab V Simpulan dan Saran

Bab ini menjelaskan mengenai simpulan dari hasil penelitian yang diperoleh, serta saran untuk penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian *Deep Reinforcement Learning* (DRL) pada penerapan jual beli saham sedang berkembang. Penelitian tersebut dilakukan dengan berbagai macam algoritma DRL dan diterapkan pada berbagai macam pasar saham. Secara garis besar, penelitian-penelitian tersebut dibagi menjadi tiga kriteria antara lain, *single algorithm* pada *single-stock trading*, *single algorithm* pada *multiple-stock trading*, dan *combine algorithms.*

State of the art dari *single algorithm on single-stock trading* menunjukkan kinerja yang bagus. Penelitian terhadap algoritma *Trading Deep Q-learning* (TDQN) yang diadopsi dari algoritma DQN menunjukkan *sharpe ratio* sebesar 1,841 untuk penerapan jual-beli saham AAPL dalam rentang waktu 01/01/2018 hingga 31/12/2019 [12]. Selain itu, perbandingan antara algoritma DQN, *Double* DQN, dan *Dueling Double* DQN telah diterapkan pada saham India untuk rentang waktu 2012 hingga 2020. Hasil dari penelitian tersebut menunjukkan bahwa *Dueling Double* DQN memiliki kinerja lebih baik daripada *Double* DQN, dan *Double* DQN menunjukkan kinerja lebih baik dari DQN [4].

Single algorithm dari DRL juga dapat diterapkan pada *multiple stock trading*. Penelitian terhadap algortima DDPG yang diterapkan pada jual beli saham di 30 Dow Jones *Stocks* menunjukkan *sharpe ratio* sebesar 1,79 untuk penerapan pada periode 01/01/2016 hingga 09/30/2018 [8]. Selain itu, algoritma A2C, PPO, DDPG menunjukkan nilai *sharpe ratio* masing-masing sebagai berikut, 1,37, 0,99, dan 0,88. Hasil tersebut diperoleh ketika diterapkan pada 30 Dow Jones *Stocks* dengan rentang waktu 2020/07/01 hingga 2022/03/31 [13]. Algoritma A2C, PPO, DDPG, TD3, dan SAC juga telah diterapkan pada Shanghai Composite Index dengan data training sebelum COVID-19, dan *data trading* setelah COVID-19. Hasil dari penelitian tersebut menunjukkan bahwa DDPG

menunjukkan kinerja terbaik dengan hasil *cumulative return* sebesar 25%, TD3 dan SAC menunjukkan hasil *cumulative return* sebesar 16% hingga 17%. Sedangkan, A2C dan PPO memiliki kinerja yang lebih rendah dari komparasi yang ada [14].

Algortima-algoritma DRL juga dapat dikombinasikan untuk melakukan pengambilan keputusan jual-beli saham. Penelitian tersebut diterapkan pada *ensemble strategy* dan *multi-agent framework*. Penerapan *ensemble trading strategy* yang menggabungkan algoritma PPO, A2C, dan DDPG dalam jual beli saham pada 30 Dow Jones *Stock* dengan rentang waktu 01/01/2016 hingga 05/08/2020 menunjukkan kinerja yang lebih baik dari masing-masing algoritma yang bekerja sendiri. *Sharpe ratio* dari *ensemble strategy,* PPO, A2C, dan DDPG masing-masing menunjukkan angka sebagai berikut, 1,3, 1,1, 1,12, dan 0,87 [7]. Selain itu, ada pula penelitian pada algoritma-algortima yang sama, dan pasar saham yang sama, namun diterapkan pada rentang waktu berbeda, yakni, 07/01/2020 hingga 06/30/2021. Masing-masing *ensemble strategy*, A2C, PPO, dan DDPG, menunjukkan nilai *sharpe ratio* sebesar, 2,81, 2,24, 2,23, dan 2,02 [15]. Penelitian *multi-agent framework* yang terdiri dari DQN *long-term* (1 jam), DQN *mid-term* (15 menit), dan DQN *short-term* (5 menit) menunjukkan hasil kinerja yang lebih baik daripada masing-masing agent tersebut yang bekerja secara terpisah. *Multi-agent framework,* DQN *long-term agent,* DQN *mid-term agent,* dan DQN *short-term agent,* masing-masing menunjukkan nilai *sharpe ratio* sebesar 0,63, 0,5, 0,41, 0,46 untuk penerapan pada data EUR/USD dengan rentang waktu 29/06/2012 hingga 25/05/2021 [9]. Kemudian terdapat juga penelitian yang mengkombinasi algoritma DQN dengan *Dueling* DQN untuk diterapkan pada pasar saham Cina dan S&P 500 dengan rentang waktu Januari 2019 hingga Januari 2021 menunjukkan hasil kinerja yang lebih bagus dari strategi lainnya (B&H *strategy*, LSTM *based agent,* TFJ-DRL *model*, HW_LSTM_RL, dan DQN) [6].

2.2. Kerangka Konseptual dan Teoritis

2.2.1. Saham

Saham merupakan surat bukti kepemilikan seseorang terhadap suatu perusahaan [16]. Saham dapat diperjual belikan di bursa saham. Transaksi tersebut diatur dan diawasi oleh pemerintah untuk melindungi investor. Harga dari saham dipengaruhi oleh permintaan dan penawaran. Jika permintaan lebih banyak dari penawaran, maka harga akan meningkat. Sebaliknya, jika penawaran lebih banyak dari permintaan, maka harga akan menurun [16]. Atribut-atribut yang dimiliki datset saham dapat dilihat pada lampiran ke sembilan[14]:

2.2.2. Analisis Teknikal

Teknikal analisis merupakan disiplin perdagangan untuk mengevaluasi dan mengidentifikasi peluang perdagangan dalam tren dan pola harga melalui grafik. Dengan kata lain, teknik ini memperhatikan perubahan harga pada periode konstan berdasarkan tampilan grafik [17]. Analisis teknikal memanfaatkan indikator teknis untuk untuk mengetahui kondisi pasar dan membantu pengambilan keputusan jual/beli. Penjelasan dan rumus dari teknikal indicator MACD, RSI, dan CCI dapat dilihat pada lampiran ke sepuluh.

2.2.3. *Deep Reinforcement Learning*

Deep Reinforcement Learning (DRL) merupakan kombinasi antara *reinforcement learning* (RL) dengan *deep learning* (DL). DRL telah mampu menyelesaikan berbagai tugas pengambilan keputusan kompleks yang sebelumnya belum mampu dijangkau oleh *machine learning* [3]. Untuk penjelasan masing-masing *deep learning* dan *reinforcement learning* dapat dilihat pada lampiran ke delapan.

A. *Deep Deterministic Policy Gradient* (DDPG)

DDPG menggabungkan *framework* dari *Q-learning* dengan *policy* gradient, dan menggunakan *neural network* sebagai fungsi pendekatan. Secara esensi, *Q-learning* merupakan metode untuk mempelajari *environment*. Namun, *Q-learning* tidak memperbarui nilai $Q(s_t, a_t)$ dengan $Q(s_{t+1}, a_{t+1})$, melainkan, *Q-learning* menggunakan *greedy action* a_{t+1} yang memaksimalkan $Q(s_{t+1}, a_{t+1})$ pada *state* s_{t+1}. Berikut rumusnya:

(1)

$$Q_{\pi}(s_t, a_t) = \mathbb{E}_{s_{t+1}}[\mathrm{r}(s_t, a_t, s_{t+1}) + \gamma \max_{a_{t+1}} \mathrm{Q}(s_{t+1}, a_{t+1})]$$

Pada setiap langkah waktu, agen DDPG melakukan *action* a_t pada s_t, kemudian agen memperoleh *reward* r_t dan tiba pada s_{t+1}. Transisi (s_t, a_t, r_t, s_{t+1}) disimpan dalam *replay buffer* R. Sejumlah N transisi diambil dari R, kemudian nilai-Q y_i diperbarui sebagai berikut:

$$\mathrm{y}_i = r_i + \gamma \mathrm{Q}'(s_{i+1}, \mu'(s_{i+1} | \theta^{\mu'}, \theta^{Q'})), \mathrm{i} = 1, \cdots, \mathrm{N} \qquad (2)$$

Kemudian jaringan *critic* diperbarui dengan meminimalkan fungsi *loss* $\mathrm{L}(\theta^Q)$ yang merupakan selisih dari keluaran jaringan *target critic* Q' dengan jaringan *critic* Q.

$$\mathrm{L}(\theta^Q) = E_{s_t, a_t, r_t, s_{t+1} \sim buffer}[(y_i - \mathrm{Q}(s_t, a_t | \theta^Q))^2] \qquad (3)$$

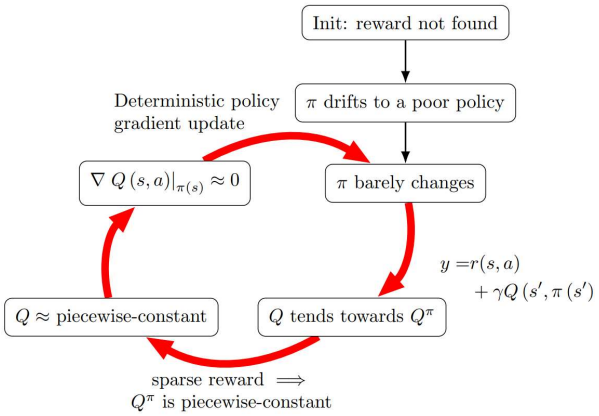
Menurut teori pada paper dengan judul *"The Problem with DDPG: Understanding Failures in Deterministic Environment s with Sparse Reward s"* (2020) menyatakan, jika algoritma DDPG memiliki *environment* pembelajarannya yang memberikan *reward* secara jarang atau ketika *environment* pembelajarannya tidak memberikan *reward* pada langkah-langkah awal, maka akan mengakibatkan *policy* yang dihasilkan *stuck* pada *local-maximal policy* atau bahkan menghasilkan *policy* yang buruk [18].

Kasus tersebut saling berkaitan. Kasus ini terjadi akibat *actor neural-network* diinisiaikan dengan nilai yang mendekati nol untuk *state-action* dan *critic neural-network* diinisiaikan dengan nilai yang mendekati nol untuk *action*, pada mulanya. Sehingga, selama agen tidak mendapatkan *reward*, maka *actor* dan *critic* dengan sangat cepat akan mengalami kondisi jenuh [18].

DDPG yang melakukan pembelajaran untuk *actor function* π_ψ (atau disebut juga *policy*) dan *critic function* Q_θ, yang direpresentasikan sebagai jaringan saraf dengan parameter *actor* ψ dan parameter *critic* θ. Pembaharuan parameter tersebut dituliskan dengan rumusan:

$$\psi \leftarrow \psi + \alpha \sum_i \frac{\partial \pi_\psi(\mathrm{si})^T}{\partial \psi} \nabla_a Q_\theta(s_i, a)|_{\mathrm{a}=\pi_\psi(s_i)}. \qquad (4)$$

Ketika *actor* bernilai ∀s, π(s) = 0.1 yang diperbarui berdasarkan $\nabla_a Q_\theta(s_i, a)|_{a=\pi_\psi(s_i)}$. Untuk $a = \pi(s) = 0.1$ gradiennya ada nol. Sehingga, *actor* tidak diperbarui. Selain itu, *critic* diperbarui dengan $y_i = r(s_i, a_i) + \gamma Q(s'_i, \pi(s'_i))$ sebagai target. Karena $Q(s'_i, 0.1)$ bernilai nol, *critic* butuh nilai selain nol untuk mendapat *reward* secara langsung, dan untuk semua sampel lainnya nilai target tetap nol. Dalam keadaan ini *critic loss* bernilai minimal, sehingga tidak ada pembaharuan lebih lanjut dari *critic*. Sehingga dihasilkanlah *policy* yang mengalami kebuntuan. Sekalinya terjebak/terhenti dalam pembaruan, maka meskipun diberikan sample yang ideal algoritma akan tetap tertahan dan tidak dapat memperbaiki *policy (deadlock)* [18]. Penjelasan tersebut digambarkan seperti grafik di bawah ini:



Gambar 2. 1 Siklus konvergensi yang mengalami deadlock [18]

B. *Proximal Policy Optimization* (PPO)

PPO merupakan algoritma yang mengontrol pembaharuan *policy* gradient dan memastikan bahwa *policy* terbarunya tidak jauh berbeda dari *policy* sebelumnya. PPO menyederhanakan objektif dari *Trust Region Policy Optimization* (TRPO) dengan menggunakan istilah clipping sebagai fungsi objektif. Berikut rumus rasio probabilitas antara *policy* sebelumnya dan *policy* terbaru:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \quad (5)$$

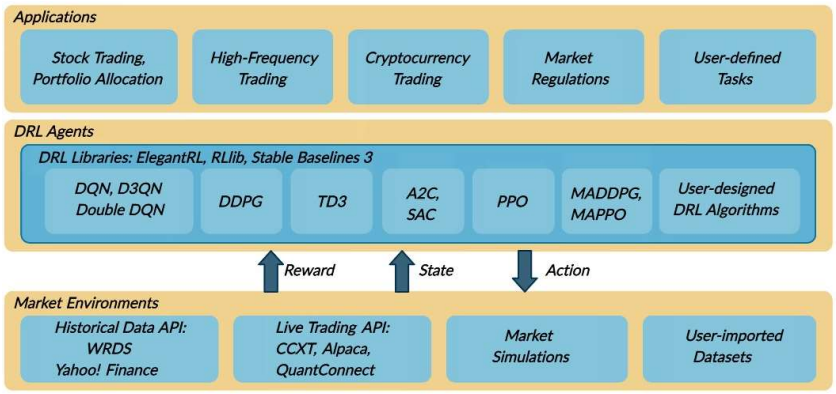
Clip pengganti fungsi objektif PPO dituliskan sebagai berikut:

$$J^{CLIP}(\theta) = E_t\left[\min\left(r_t(\theta)A(s_t, a_t), Clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A(s_t, a_t)\right)\right] \quad (6)$$

dengan $r_t(\theta)A(s_t, a_t)$ merupakan normal *policy* gradient objective. $A(s_t, a_t)$ merupakan estimasi advantage function. Fungsi $Clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon)$ melakukan klip rasio $r_t(\theta)$ di dalam $[1 - \epsilon, 1 + \epsilon]$. Fungsi objektif dari PPO mengambil nilai minimum dari objektif nilai *clip* dan nilai normal. PPO mencegah perubahan *policy* yang besar bahkan bergerak keluar luar *clipped rentang*. Oleh karena itu, PPO meningkatkan stabilitas kebijakan pelatihan jaringan dengan membatasi pembaruan kebijakan di setiap langkah pelatihan.

Sebuah teori menyatakan bahwa pada algoritma PPO, ketika sebuah agen memiliki jumlah interaksi yang terbatas dengan *environment*, maka mekanisme tersebut akan mendorong pada *policy* yang *local maximal* [20][21].

2.2.4. Pustaka FinRL

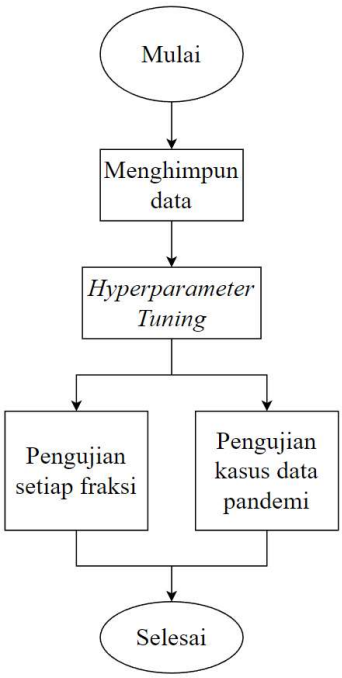


Gambar 2. 2 Lapisan pustakan *Financial Reinforcement Learning* [5]

FinRL merupakan *library* yang digunakan untuk melakukan *trading*. Pustaka tersebut terdiri dari tiga lapisan yang terdiri dari applications, DRL agents, dan market environment. Penjelasan dari masing-masing lapisan bisa ditinjau pada lampiran ke sebelas.

BAB III
PERANCANGAN SISTEM

3.1. Desain Sistem



Gambar 3. 1 Diagram Alur Penelitian Secara Umum

Berikut penjelasan masing-masing proses dari diagram alur penelitian secara umum tersebut:

3.1.1. Menghimpun Data

Penghimpunan data ditujukan sebagai landasan inisiasi variabel pada program. Penghimpunan data dilakukan untuk beberapa aspek seperti:

1) Menghimpun data *hyperparameter* apa saja yang digunakan untuk masing-masing algoritma dari penelitian sebelumnya, dan berapa nilai yang diinisiasikan untuk setiap *hyperparameter* berdasarkan pada penelitian sebelumnya.

Hyperparameter dan nilai *hyperparameter* yang diperoleh dari hasil penelitian sebelumnya untuk masing-masing algoritma adalah sebagai berikut:

a) Algoritma PPO

Tabel 3. 1 *Hyperparameter* PPO yang dijadikan rujukan

n_steps	ent_coef	learning_rate	batch_size
50000	0.01	0.00025	128

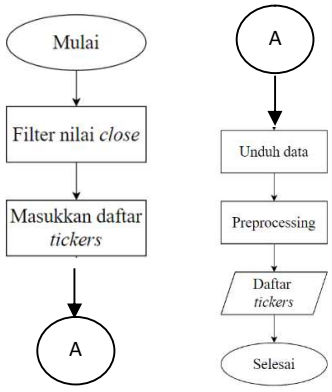
b) Algoritma DDPG

Tabel 3. 2 *Hyperparameter* DDPG yang dijadikan rujukan

learning_rate	batch_size	buffer_size
0.001	128	50000

2) Menghimpun daftar data saham-saham akan diuji pada penelitian ini. Terdapat dua skenario untuk pemilihan data saham yang akan diimplemplementasikan pada pengujian. Berikut skenarionya:

a) Untuk pengujian stabilitas masing-masing algoritma terhadap kelima fraksi saham di Indonesia, maka pemilihan daftar saham dilakukan dengan merujuk pada data idx.co.id yang diakses pada tanggal 27/08/2022. Berikut alur pemilihan daftar data saham masing-masing fraksi:



Gambar 3. 2 Alur pendataan daftar saham untuk pengujian stabilitas algoritma pada kelima fraksi

Pada data saham yang diperoleh dari idx.co.id, terdapat banyak kolom, termasuk kolom *"stock code",* dan *"close"*. Tahap pertama yang dilakukan yaitu menyaring data berdasarkan kolom *"close"* yang memenuhi rentang harga masing-masing fraksi. Rentang harga saham dari masing-masing fraksi adalah sebagai berikut:

i) Fraksi 1, $x < Rp200{,}00$
ii) Fraksi 2, $\text{Rp}200{,}00 \leq x < Rp500{,}00$
iii) Fraksi 3, $\text{Rp}500{,}00 \leq x < Rp2000{,}00$
iv) Fraksi 4, $\text{Rp}2000{,}00 \leq x < Rp5000{,}00$
v) Fraksi 5, $x \geq Rp5000{,}00$

Keterangan:

x: harga per lembar dari suatu saham

Setelah mendapatkan daftar kode saham untuk masing-masing fraksi, daftar kode setiap fraksi dimasukkan ke dalam program untuk diunduh dengan rentang tanggal 2009/01/01 hingga 20/19/12/20, kemudian dilakukan preprocessing. Salah satu proses dalam preprocessing tersebut yaitu membersihkan data mentah yang memiliki data-data yang tidak lengkap dengan cara menghapus saham tersebut dari daftar. Sehingga data saham yang tidak lengkap tidak dimasukkan ke dalam proses *trading*. Dari proses tersebut, dihasilkan daftar data saham yang dapat diproses untuk penelitian. Masing-masing fraksi diambil empat data saham. Berikut daftar saham yang digunakan:

i) Fraksi 1 terdiri dari KIJA, LCGP, LMPI, LPKR;
ii) Fraksi 2 terdiri dari BMTR, BTON, FORU, GEMA;
iii) Fraksi 3 terdiri dari AKRA, BRPT, KLBF, MEDC;
iv) Fraksi 4 terdiri dari JECC, TMAS, TPIA, UNVR;
v) Fraksi 5 terdiri dari INCO, INDF, INTP, UNTR;

b) Untuk pengujian stabilitas masing-masing algoritma terhadap pengaruh data saat pandemi di Indonesia, maka, daftar saham yang digunakan adalah BRPT, KLBF, SCMA, dan UNVR. Daftar data saham tersebut

diperoleh dari seleksi *preprocessing* dari 30 daftar saham JII *(Jakarta Islamic Index)*.

3) Menghimpun data terkait rentang waktu *dataset* yang digunakan dalam penelitian, baik itu rentang waktu *dataset* secara keseluruhan maupun rentang waktu *dataset* yang terdampak COVID-19.

Dalam menentukan rentang waktu *dataset* keseluruhan (rentang waktu data training dan rentang waktu *data trading*) yang digunakan untuk penelitian ini, maka angka yang digunakan berlandaskan informasi dari penelitian sebelumnya. Berikut merupakan beberapa penelitian yang dijadikan rujukan:

Tabel 3. 3 Rujukan untuk menentukan rentang waktu *dataset* secara keseluruhan

Sumber Literasi	Rentang Waktu Data Penelitian
Deep Reinforcement Learning for Automated *Stock trading*: An *Ensemble strategy* oleh Hongyang Yang, dkk. (2020)	2009/01/01 hingga 2020/05/08
Practical *Deep Reinforcement Learning* Approach for *Stock trading* oleh Hongyang Yang, dkk. (2022)	2009/01/01 hingga 2018/09/30
FinRL: A *Deep Reinforcement Learning Library* for Automated *Stock trading* in Quantitative Finance oleh Hongyang Yang, dkk. (2022)	2009/01/01 hingga 2020/09/23

Berdasarkan data di atas, maka, penelitian ini juga menggunakan *dataset* yang besar untuk melakukan penelitian, yakni dengan rentang waktu 2009/01/01 hingga 2022/08/15.

Selain itu, dibutuhkan pula informasi perihal rentang waktu pandemi COVID-19 yang berdampak pada saham Indonesia. Informasi tersebut