

# Perdiksi Harga Minyak Mentah Menggunakan Model Prophet dan Evaluasi Algoritma Monte Carlo pada Lingkungan Reinforcement Learning

Reni Kartika Suwandi

Program Studi Informatika

Fakultas Teknologi Informasi

Universitas Sebelas April, Kabupaten Sumedang

Email: [220660121130@student.unsap.ac.id](mailto:220660121130@student.unsap.ac.id)

---

## ABSTRACT

*Global oil price fluctuations significantly impact various economic sectors, from industries to macroeconomic conditions. Accurate oil price predictions are crucial for strategic decision-making across sectors. This study leverages the Prophet model, developed by Facebook, for its robust time series forecasting capabilities. The Prophet model handles trend, seasonality, and holiday effects effectively, making it suitable for predicting oil prices. Additionally, reinforcement learning (RL) algorithms, particularly Monte Carlo control, demonstrate potential in dynamic decision-making environments. This research aims to integrate Prophet's forecasting with Monte Carlo control in RL to enhance prediction accuracy and decision-making efficiency. The study involves simulating agent decisions in a dynamic environment using the CartPole-v1 setup from OpenAI Gym. The findings indicate that combining Prophet and Monte Carlo control provides deeper insights into oil price movements and decision-making strategies. This integrated approach can improve the accuracy of oil price forecasts and enhance decision-making processes in energy-related sectors.*

---

**Keywords** - Oil Price Prediction, Prophet Model, Monte Carlo Control, Reinforcement Learning

---

## 1. Pendahuluan

Harga minyak dunia memiliki dampak yang signifikan pada berbagai sektor ekonomi, mulai dari industri hingga ekonomi makro secara keseluruhan. Fluktuasi harga minyak yang tidak terduga dapat mengakibatkan konsekuensi yang luas, baik pada tingkat individu maupun skala nasional. Oleh karena itu, kemampuan untuk memprediksi pergerakan harga minyak secara akurat menjadi krusial untuk pengambilan keputusan strategis di berbagai sektor.

Metode prediksi harga minyak telah menjadi fokus penelitian intensif selama bertahun-tahun. Salah satu pendekatan terbaru yang menunjukkan potensi besar adalah penggunaan model Prophet. Prophet, dikembangkan oleh Facebook, adalah alat prediksi time series yang mampu menangani komponen tren, musiman, dan hari libur dalam data dengan akurat. Kemampuan Prophet untuk mengakomodasi data time series yang kompleks menjadikannya pilihan yang tepat untuk memprediksi harga minyak.

Di sisi lain, kecerdasan buatan (AI) dan pembelajaran mesin (ML) telah menunjukkan bahwa algoritma reinforcement learning (RL), seperti Monte Carlo control, dapat digunakan untuk mengembangkan agen yang mampu membuat keputusan dalam lingkungan yang dinamis dan kompleks. Kombinasi prediksi menggunakan Prophet dengan evaluasi kinerja Monte Carlo control

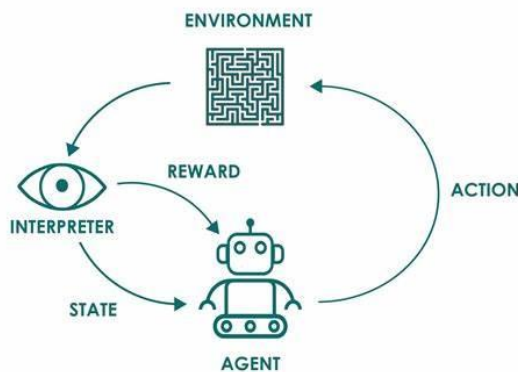
dalam RL diharapkan dapat meningkatkan akurasi dan efisiensi pengambilan keputusan otomatis dalam berbagai aplikasi praktis.

Studi ini bertujuan untuk mengintegrasikan dua pendekatan analitis yang berbeda ini untuk meningkatkan prediksi harga minyak dan efektivitas pengambilan keputusan, memberikan wawasan yang lebih dalam bagi peneliti dan praktisi di bidang ini.

## 2. Landasan Teori

### 2.1 Agen dan Lingkungan

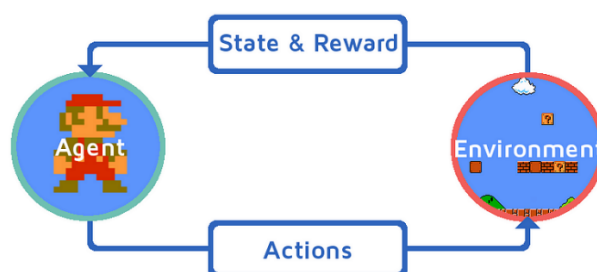
Lingkungan dan agen ibarat ruang dan objek yang keduanya sangat penting untuk berlangsungnya ekosistem siklus hidup dalam penyelesaian suatu permasalahan. Apapun kasusnya, dapat digeneralisasi dalam bentuk sebuah peta yang dapat mengilustrasikan hubungan dari keduanya maupun bagian-bagian yang ada di antara keduanya. Abstraksi pada Gambar 1 merupakan model cara komputasi sebuah agen, antara tindakan yang akan dikerjakan sesuai kondisi lingkungannya (Russell, 2003).



**Gambar 1.** Agent dan Lingkungan

### 2.2 Jenis Lingkungan dan penerapannya

Lingkungan dalam penerapan machine learning sangat luas. Penggunaan model Prophet untuk memprediksi harga minyak dan penerapan metode Monte Carlo control dalam reinforcement learning untuk mengoptimalkan pengambilan keputusan agen di lingkungan simulasi. Lingkungan simulasi ini menggunakan lingkungan "CartPole-v1" dari OpenAI Gym, yang merupakan masalah kontrol klasik di mana agen harus menyeimbangkan tiang pada gerobak yang bergerak.<sup>1</sup>



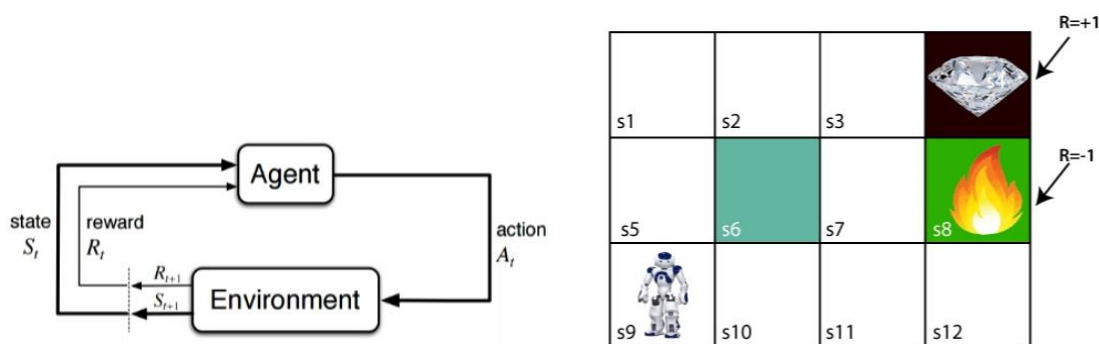
**Gambar 2.** Contoh Penerapan Agent di lingkungan CartPole OpenAI Gym pada game Mario

<sup>1</sup> Setyawan, G. E. (2019). Pengembangan Reinforcement Learning (RL) Single-Agent Menggunakan Improve Q-Learning. [https://www.researchgate.net/publication/338171196\\_dshd](https://www.researchgate.net/publication/338171196_dshd)

### 2.3 Reinforcement Learning

Reinforcement learning adalah jenis pembelajaran mesin di mana agen belajar membuat keputusan dengan mencoba berbagai tindakan dalam suatu konteks dan menerima umpan balik dalam bentuk penghargaan (penghargaan) atau penalti. Proses ini sebanding dengan cara manusia belajar dari coba-coba. Agen secara aktif mencari pengalaman baru untuk meningkatkan pemahaman dan kinerja mereka sambil mengandalkan data yang ada. Agen memiliki tujuan untuk memaksimalkan jumlah reward yang diterima dalam jangka waktu tertentu.<sup>2</sup>

Dalam situasi di mana keputusan harus dibuat secara berurutan, seperti navigasi, pengendalian robot, atau permainan, metode ini sangat berguna. Misalnya, lengan robot dapat belajar memanipulasi objek dengan benar dengan terus mencoba dan menerima umpan balik sampai gerakan yang diinginkan tercapai melalui proses pembelajaran penguatan.



**Gambar 3.** Penerapan Reinforcement Learning

### 2.4 Pengertian Metode Monte Carlo

Simulasi Monte Carlo ditemukan sebelum munculnya komputer modern - metode ini dikembangkan selama Perang Dunia II - oleh dua ahli matematika: Stanisław Ulam dan John von Neumann. Pada saat itu, keduanya terlibat dalam proyek Manhattan, dan mereka menciptakan teknik ini untuk mensimulasikan reaksi berantai dalam uranium yang sangat diperkaya. Secara sederhana, mereka sedang mensimulasikan ledakan atom<sup>3</sup>.

Metode Monte Carlo adalah teknik yang digunakan dalam pemecahan masalah Reinforcement Learning (RL) dengan cara mengestimasi nilai fungsi berdasarkan rata-rata pengembalian sampel. Dalam konteks ini, pengembalian merujuk pada keuntungan kumulatif yang diharapkan di masa depan. Untuk memastikan pengembalian yang terdefinisi dengan baik, metode Monte Carlo biasanya diterapkan pada tugas-tugas episodik, yaitu tugas-tugas yang memiliki titik awal dan titik akhir yang jelas.

Metode ini mengandalkan pengumpulan data dari serangkaian episode di mana setiap episode mencakup perjalanan atau langkah-langkah yang diambil sesuai dengan kebijakan tertentu. Setelah itu, nilai fungsi dihitung dengan mengambil rata-rata dari semua pengembalian yang diamati dari kunjungan ke negara bagian tersebut.

<sup>2</sup> Richard S.Sutton., Barto.A.G.(2018). Reinforcement Learning An Introduction, Second Edition.Cambridge, Massachusetts. : The MIT Press.

<sup>3</sup> Richard S.Sutton., Barto.A.G.(2018). Reinforcement Learning An Introduction, Second Edition.Cambridge, Massachusetts. : The MIT Press

### 3. Metode Penelitian

#### 3.1 Model Prophet

Model Prophet adalah sebuah alat yang digunakan untuk melakukan peramalan terhadap data time series. Model ini dikembangkan oleh Facebook dan dirancang untuk memfasilitasi peramalan yang akurat dengan menangani karakteristik unik dari data time series. Prophet sangat efektif dalam menangkap tren musiman dan fluktuasi yang terjadi pada data historis.

Salah satu aplikasi nyata dari model ini adalah untuk memprediksi harga minyak mentah di masa depan. Prophet mampu memberikan prediksi yang mempertimbangkan pola musiman dan variasi harga yang terjadi secara keseluruhan.

Implementasi Model Prophet pada Studi Kasus
<pre># Membuat model Prophet model = Prophet()  # Fitting model with data model.fit(co_data[['ds', 'y']]) # Melakukan prediksi menggunakan model Prophet future = model.make_future_dataframe(periods=24, freq='M') forecast = model.predict(future)  # Pastikan kolom 'yhat' ada dalam hasil prediksi if 'yhat' in forecast.columns:     # Plot ramalan     fig1 = model.plot(forecast)     plt.title('')     plt.show()      # Tampilkan hasil prediksi terakhir     print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail(12)) else:     print("Kolom 'yhat' tidak ditemukan dalam hasil prediksi. Kolom yang tersedia:", forecast.columns)</pre>

**Tabel 1.** Pseudocode Model Prophet pada Studi Kasus

#### 2.5 Prediksi Monte carlo

Metode Monte Carlo (MC) digunakan untuk memperkirakan nilai suatu state bagian dalam kebijakan tertentu dengan cara mengamati dan rata-rata keuntungan kumulatif yang diharapkan dari kunjungan ke state bagian tersebut. menggunakan metode MC, semakin banyak keuntungan yang diamati dari kunjungan ke suatu negara bagian, semakin mendekati rata-rata keuntungan tersebut terhadap nilai yang diharapkan. Hal ini membuat metode MC efektif dalam mempelajari dan memprediksi nilai-nilai negara bagian dalam kebijakan tertentu berdasarkan pengalaman yang diamati.<sup>4</sup>

Cara untuk menganalisis prediksi monte carlo:

<sup>4</sup> Richard S.Sutton., Barto.A.G.(2018). Reinforcement Learning An Introduction, Second Edition.Cambridge, Massachusetts. : The MIT Press

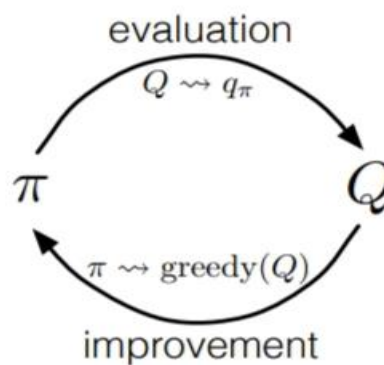
Prediksi MC kunjungan pertama, untuk memperkirakan $V^{\pi}$
Input: sebuah kebijakan $\pi$ untuk dievaluasi Inisialisasi: $V(s) \leftarrow R$ , sembarang, untuk semua $s \in S$ Returns(s) sebuah daftar kosong, untuk semua $s \in S$ Loop selamanya (untuk setiap episode): Buat sebuah episode mengikuti $\pi$ : $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_T$ 2.6 Kontrol Monte Carlo

Tabel 2. Pseudocode MC kunjungan pertama

## 2.5 Pengendalian Monte Carlo

Metode Monte Carlo dapat digunakan dalam pengendalian untuk memperkirakan kebijakan yang optimal. Ide dasarnya sejalan dengan konsep Generalized Policy Iteration (GPI). Dalam GPI, pendekatan yang digunakan adalah mempertahankan dan memperbaiki secara bergantian kebijakan dan estimasi fungsi nilai. Fungsi nilai diperbarui secara berulang untuk mendekati fungsi nilai dari kebijakan saat ini, sementara kebijakan diperbaiki berdasarkan estimasi fungsi nilai saat ini.<sup>5</sup>

Proses ini menciptakan dinamika yang saling mempengaruhi antara kebijakan dan fungsi nilai, di mana perubahan dalam satu komponen menciptakan target untuk perbaikan yang berkelanjutan dalam komponen lainnya. Dengan menggabungkan perubahan ini, baik fungsi kebijakan maupun nilai dapat mendekati kondisi optimal secara bertahap.



Gambar 4. Pengendalian Monte Carlo

Untuk memulai, mari pertimbangkan iterasi kebijakan klasik versi Monte Carlo. Dengan metode ini, kita melakukan langkah-langkah evaluasi kebijakan dan perbaikan kebijakan secara lengkap secara bergantian, dimulai dari kebijakan sewenang-wenang  $\pi_0$  dan berakhir dengan kebijakan optimal serta fungsi nilai tindakan optimal

$$\pi_0 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} q_*$$

<sup>5</sup> Richard S.Sutton., Barto.A.G.(2018). Reinforcement Learning An Introduction, Second Edition.Cambridge, Massachusetts. : The MIT Press

Dimana  $\xrightarrow{E}$  menunjukkan evaluasi kebijakan yang lengkap  $\xrightarrow{I}$  menunjukkan peningkatan kebijakan yang lengkap. Evaluasi kebijakan dilakukan dengan tepat seperti yang dijelaskan pada bagian sebelumnya. Banyak episode dialami, dengan fungsi nilai aksi perkiraan mendekati fungsi sejati secara asimtotik. Di bawah asumsi-asumsi ini, metode Monte Carlo akan menghitung setiap  $q_{\pi^*}$  dengan tepat, untuk  $\pi_k$  sembarang.

Dalam kasus ini, kita memiliki fungsi nilai aksi, dan oleh karena itu tidak diperlukan model untuk membangun kebijakan rakus. Untuk setiap fungsi nilai aksi  $q$ , kebijakan rakus yang sesuai adalah yang, untuk setiap  $s \in S$ , secara deterministik memilih tindakan dengan nilai aksi maksimal:

$$\pi(s) = \arg \max_a q(s, a)$$

Peningkatan kebijakan dapat dilakukan dengan membangun setiap  $\pi_{k+1}$  sebagai kebijakan rakus terhadap  $q_{\pi_k}$ . Teorema peningkatan kebijakan kemudian berlaku untuk  $\pi_k$ . Dan  $\pi_{k+1}$  karena untuk semua  $s \in S$

$$\begin{aligned} q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg \max_a q_{\pi_k}(s, a)) \\ &= \max_a q_{\pi_k}(s, a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \\ &\geq v_{\pi_k}(s) \end{aligned}$$

Teorema menjamin bahwa setiap  $\pi_{k+1}$  secara seragam lebih baik dari  $\pi_k$ , atau setidaknya sama baik dengan  $\pi_k$ , dalam hal ini keduanya adalah kebijakan optimal. Ini pada gilirannya menjamin bahwa proses secara keseluruhan konvergen ke kebijakan optimal dan fungsi nilai optimal. Dengan cara ini, metode Monte Carlo dapat digunakan untuk menemukan kebijakan optimal hanya dengan menggunakan episode sampel tanpa pengetahuan lain tentang dinamika lingkungan.

#### Implementasi Pengendalian Monte Carlo Pada Studi Kasus

```
# Monte Carlo Kontrol
for i_episode in range(1, n_episodes + 1):
    state = tuple(env.reset())
    episode = []
    epsilon = max(epsilon_end, epsilon_start *
        (epsilon_decay ** i_episode))
    total_reward = 0
    for t in range(max_t):
        if state not in Q:
            Q[state] = {a: 0.0 for a in range(action_size)}

    # Dapatkan prediksi prophet untuk langkah waktu berikutnya
    next_time_idx = len(co_data) + t # dataframe
    Indeks untuk langkah waktu berikutnya di masa depan
    if next_time_idx < len(forecast):
        predicted_price =
forecast['yhat'].iloc[next_time_idx]
# Anda juga dapat menggunakan batas ketidakpastian jika
diperlukan: predicted_price =
forecast['yhat_lower'].iloc[next_time_idx]
```

```

    prophet        # Lakukan tindakan RL berdasarkan prediksi
    berdasarkan predicted_price
    reward = calculate_reward(predicted_price)

    action = get_action(state, epsilon)
    next_state, _, done, _ = env.step(action)
    episode.append((state, action, reward))
    total_reward += reward

    if done:
        break

    state = tuple(next_state)
else:
    break # Berhenti jika next_time_idx melebihi
panjang perkiraan

total_rewards.append(total_reward)

# Perbarui nilai Q menggunakan metode Monte Carlo
G = 0
for state, action, reward in reversed(episode):
    if state not in Q:
        Q[state] = {a: 0.0 for a in range(action_size)}

    G = gamma * G + reward
    if not (state, action) in [(x[0], x[1]) for x in
episode[:-1]]:
        if (state, action) not in returns_sum:
            returns_sum[(state, action)] = 0.0
            returns_count[(state, action)] = 0
        returns_sum[(state, action)] += G
        returns_count[(state, action)] += 1
        Q[state][action] = returns_sum[(state, action)]
/ returns_count[(state, action)]

    if i_episode % 100 == 0:
        print(f"Episode {i_episode}/{n_episodes}")

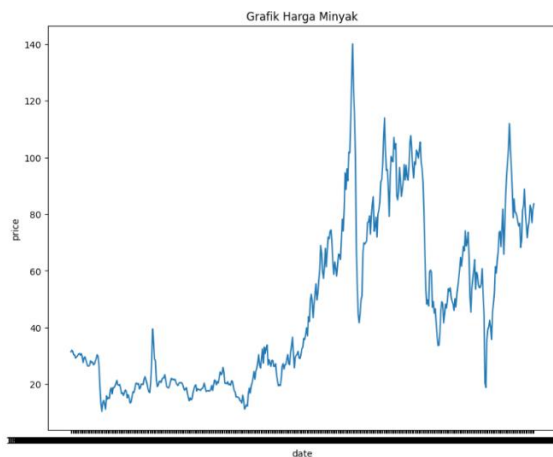
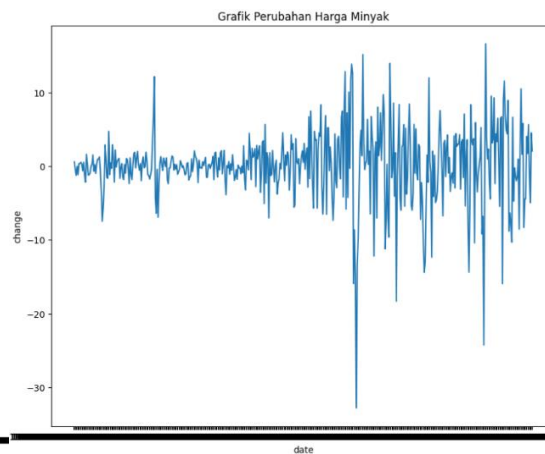
env.close()

```

**Tabel 3.** Pseudocode Pengendalian Monte Carlo

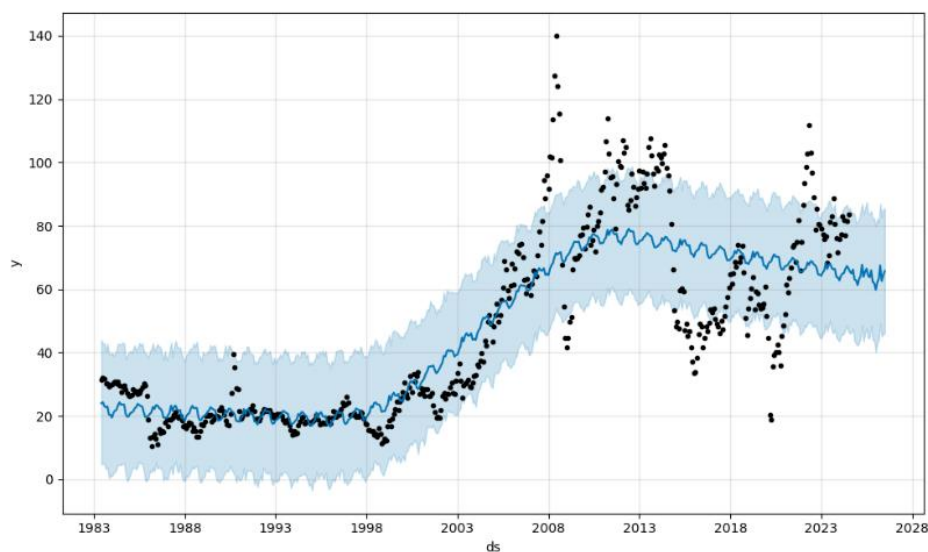
#### 4. Hasil dan Analisis

Penelitian ini menggunakan model Prophet dan metode Monte Carlo control menghasilkan wawasan yang relevan, mencakup prediksi harga minyak dan simulasi pengambilan keputusan. Data yang diperoleh memberikan pemahaman mendalam terhadap pergerakan harga serta evaluasi strategi keputusan dalam lingkungan yang dinamis.

**Gambar 5.** Grafik Harga Minyak**Gambar 6.** Grafik Perubahan Harga minyak

Hasil dari grafik gambar 5 menunjukkan fluktuasi dari harga minyak mentah, sedangkan Gambar 6 menunjukkan Grafik Perubahan harga minyak penelitian menunjukkan bahwa harga minyak mentah mengalami tren yang naik dan turun selama periode waktu yang ditunjukkan.

Untuk memprediksi harga minyak mentah di masa mendatang, data historis akan diuji menggunakan model Prophet. Model ini dirancang untuk menangkap tren musiman dan fluktuasi harga secara keseluruhan, memungkinkan kita untuk mendapatkan perkiraan yang lebih akurat mengenai perubahan harga di masa depan.

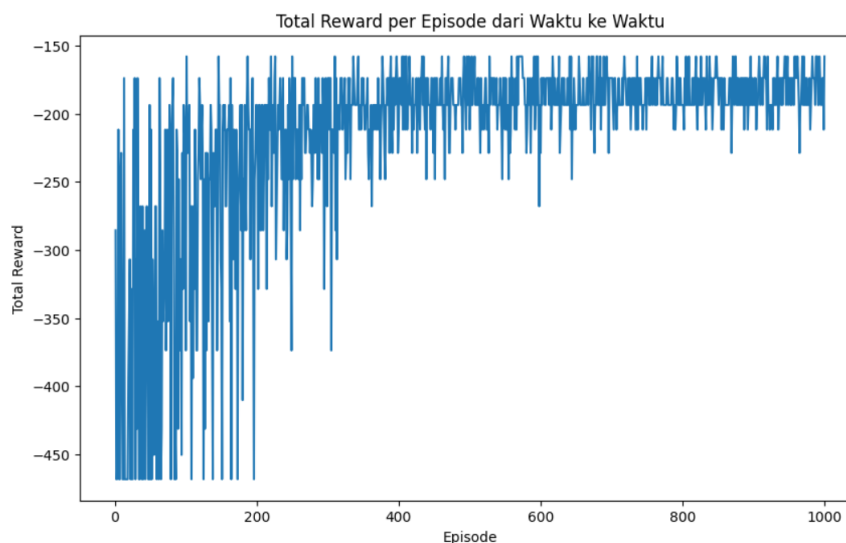
**Gambar 7.** Grafik Forecaster Prediksi Harga Minyak Mentah

Gambar 7 menunjukkan grafik prediksi harga minyak mentah berdasarkan data historis yang dimodelkan menggunakan Prophet. Grafik ini memberikan gambaran visual tentang pergerakan harga minyak mentah selama periode waktu tertentu. Sumbu Y menunjukkan harga dalam dolar AS per barel, sementara sumbu X menunjukkan waktu.

Hasil tersebut memberikan prediksi harga minyak mentah untuk periode waktu tertentu di masa depan. Tren yang terlihat dalam grafik ini menunjukkan bagaimana harga minyak mentah diproyeksikan bergerak, memungkinkan para pemangku kepentingan untuk membuat keputusan yang lebih informasi dalam konteks volatilitas pasar energi. Setelah melakukan prediksi harga



minyak mentah menggunakan model Prophet, data yang sudah dimodelkan tersebut akan disimulasikan menggunakan metode pengendalian Monte Carlo



**Gambar 8.** Total Reward Per Episode

Grafik gambar 8 menunjukkan kecenderungan umum perubahan total reward per episode, dengan beberapa fluktuasi. Hal ini menunjukkan bahwa agen sedang belajar dan memperbaiki kinerjanya dari waktu ke waktu. Namun, fluktuasi dalam total reward per episode menunjukkan bahwa agen masih dalam tahap eksplorasi dan pembelajaran. Lonjakan dalam total reward per episode mungkin disebabkan oleh agen menemukan strategi baru yang lebih efektif.

Dengan demikian prediksi harga minyak mentah, terindikasi bahwa prediksi harga akan terus berfluktuasi dengan berbagai faktor, mirip dengan bagaimana agen mengalami fluktuasi dalam total rewardnya seiring waktu.

## 5. Kesimpulan

Penelitian ini mengintegrasikan dua pendekatan analitis yang berbeda untuk meningkatkan prediksi harga minyak dan efektivitas pengambilan keputusan. Pendekatan pertama menggunakan model Prophet, yang efektif dalam menangkap tren musiman dan fluktuasi harga berdasarkan data time series historis. Prophet memberikan perkiraan yang akurat untuk pergerakan harga di masa depan, memungkinkan pengambil keputusan untuk merencanakan strategi yang lebih baik dalam menghadapi volatilitas pasar energi.

Pendekatan kedua melibatkan penggunaan metode Monte Carlo dalam reinforcement learning (RL), di mana simulasi digunakan untuk mengoptimalkan kebijakan agen dalam lingkungan yang dinamis dan kompleks. Evaluasi kinerja agen menggunakan Monte Carlo control menunjukkan bahwa agen mampu belajar dan meningkatkan kinerjanya seiring waktu, meskipun mengalami fluktuasi selama proses eksplorasi.

Hasil dari penelitian ini menunjukkan bahwa integrasi Prophet dan Monte Carlo control dapat memberikan wawasan yang lebih dalam tentang pergerakan harga minyak serta strategi pengambilan keputusan yang lebih efektif. Kombinasi dua pendekatan ini dapat digunakan tidak hanya untuk meramalkan harga minyak dengan lebih akurat, tetapi juga untuk meningkatkan keefektifan pengambilan keputusan di berbagai sektor yang terkait dengan pasar energi global.

### Referensi

- [1] Setyawan, G. E. (2019). *Pengembangan Reinforcement Learning (RL) Single-Agent Menggunakan Improve Q-Learning*. <https://www.researchgate.net/publication/338171196>
- [2] Richard S.Sutton., Barto.A.G.(2018). *Reinforcement Learning An Introduction*, Second Edition.Cambridge, Massachusetts. : The MIT Press