Chapter 18: Summary

Diedrick Darrell Darmadi - 1103223031

Reinforcement learning berkembang menjadi salah satu bidang paling menarik dalam kecerdasan buatan karena menggabungkan pengambilan keputusan, pembelajaran berbasis pengalaman, serta optimasi hasil jangka panjang. Inti konsepnya adalah bahwa seorang agen mengamati kondisi lingkungan, memilih tindakan tertentu, lalu menerima hadiah atau hukuman. Melalui proses ini, agen secara bertahap mempelajari strategi yang memaksimalkan total reward dalam jangka panjang. Pendekatan ini menyerupai proses belajar manusia, di mana perilaku dibentuk oleh konsekuensi positif dan negatif. Berbagai aplikasinya mencakup robotika, permainan papan, sistem pengatur suhu otomatis, perdagangan saham, hingga game komputer. Gambar 1 menunjukkan sejumlah contoh tersebut, mulai dari navigasi robot, permainan Atari seperti Ms. Pac-Man, pertandingan Go, pengaturan energi, hingga sistem trading otomatis.

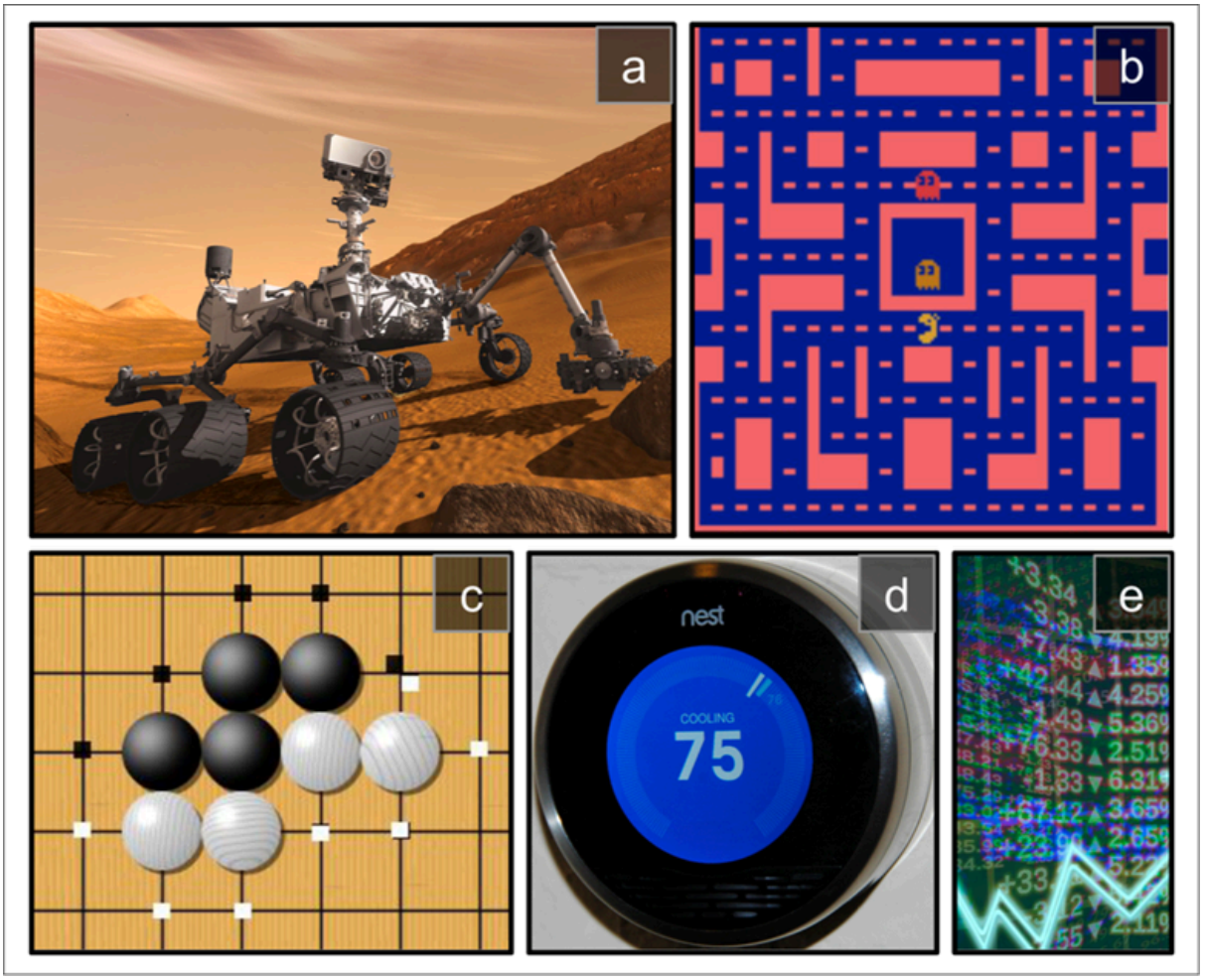
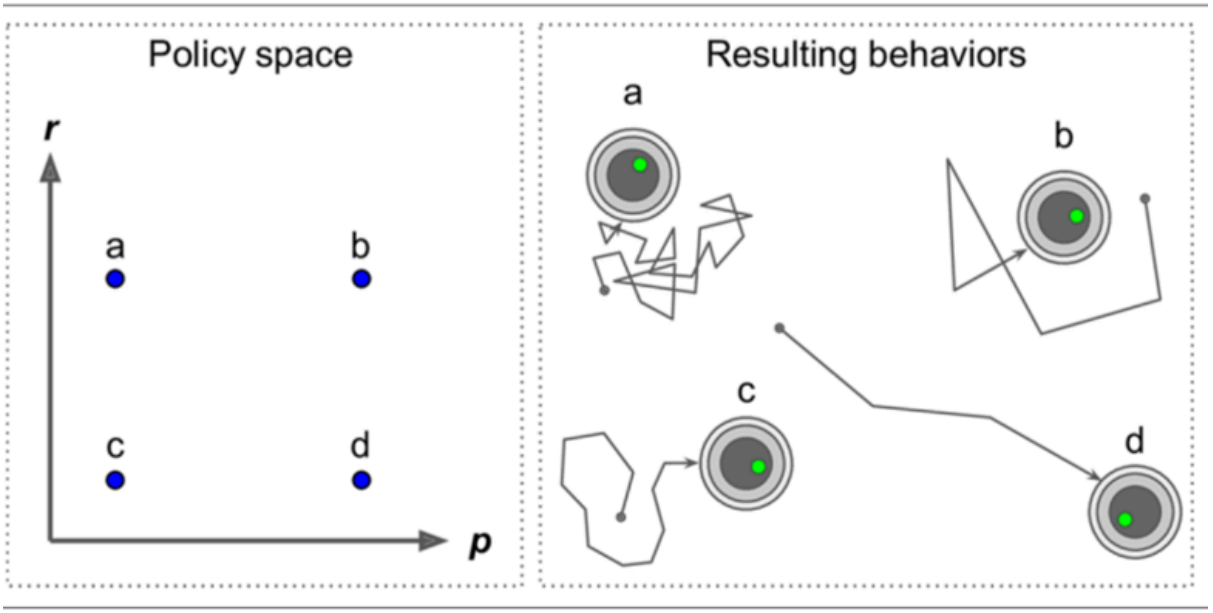


Figure 1 Reinforcement Learning examples

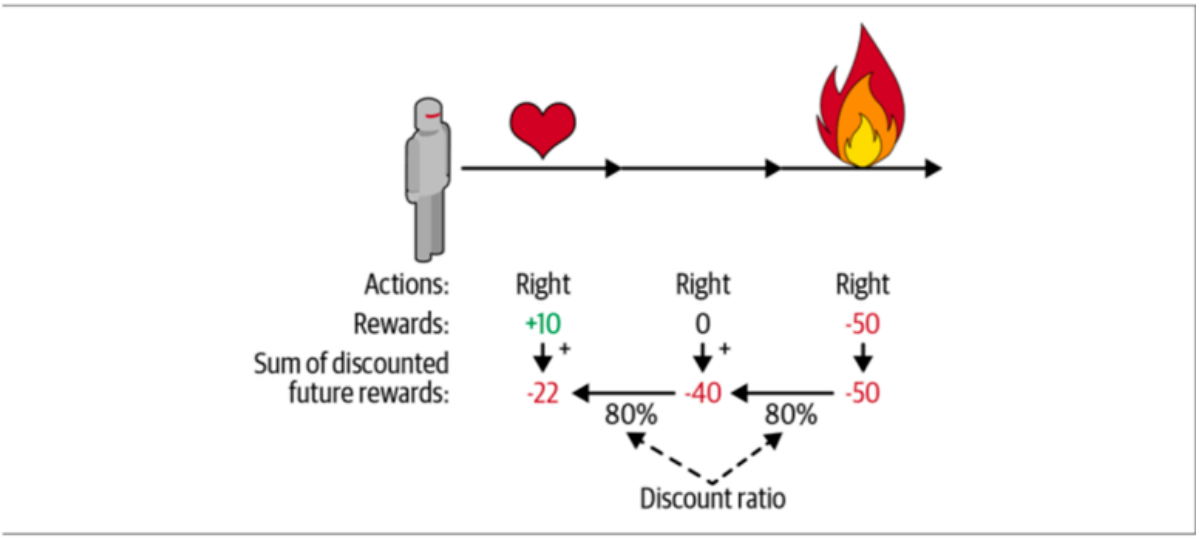
Langkah pertama dalam merancang agen adalah menentukan **kebijakan** (policy), yaitu aturan yang menghubungkan observasi dengan tindakan. Kebijakan bisa bersifat deterministik maupun acak. Misalnya, vacuum robot dapat diarahkan untuk bergerak maju dengan peluang p atau berputar acak dengan peluang 1−p. Unsur stokastik membantu agen menjelajahi berbagai kemungkinan. Untuk

melatih kebijakan, dapat digunakan metode pencarian brute-force, algoritma genetika (Gambar 2), atau pendekatan berbasis gradien seperti *policy gradients*. Metode berbasis gradien menyesuaikan parameter kebijakan menuju arah yang diperkirakan meningkatkan reward jangka panjang.

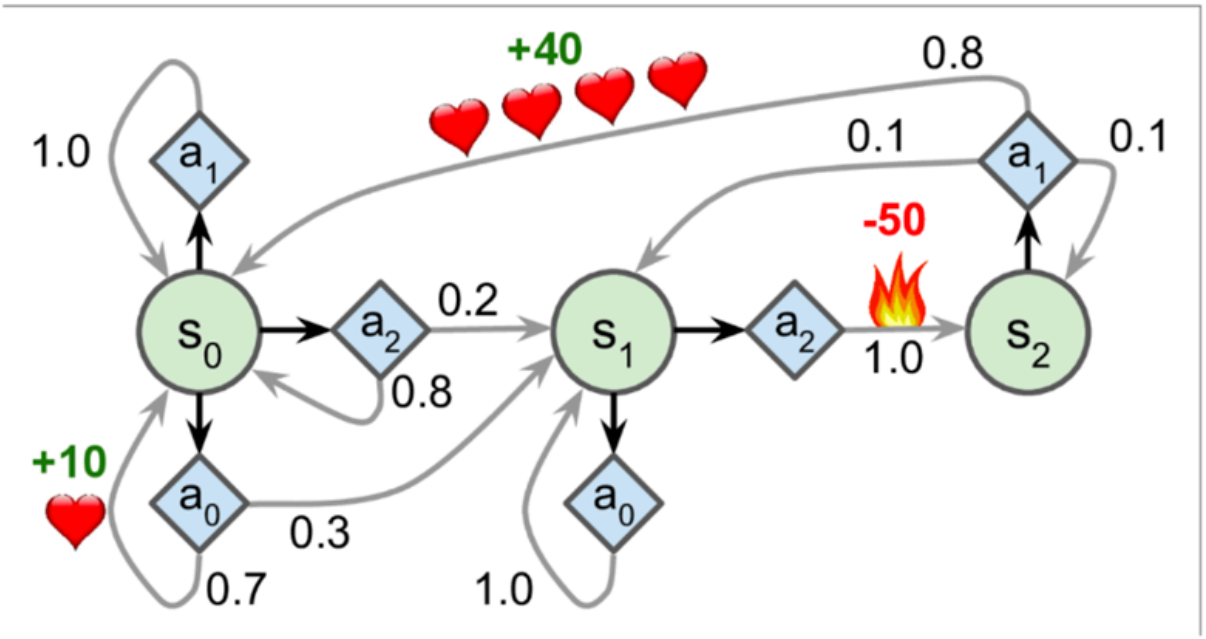


Untuk melakukan eksperimen, lingkungan simulasi memegang peranan penting. OpenAI Gym menyediakan berbagai jenis lingkungan, mulai dari simulasi fisika sederhana seperti CartPole hingga permainan Atari dan robotika 3D. Pada tugas CartPole, agen menerima empat variabel keadaan—posisi dan kecepatan kart, sudut pole, serta kecepatan sudutnya—lalu harus memilih apakah mendorong kart ke kiri atau ke kanan. Kebijakan berbasis aturan sederhana biasanya hanya berhasil sebagian, sementara kebijakan dengan jaringan saraf (neural network policy) mampu memberikan performa jauh lebih baik.

Pelatihan agen memunculkan tantangan besar terkait **credit assignment problem**, yaitu menentukan tindakan mana yang sebenarnya berkontribusi pada reward yang muncul kemudian. Untuk mengatasinya, setiap aksi diberikan nilai *return*, yaitu jumlah reward masa depan yang didiskon (Gambar 3)Faktor diskon—misalnya 0,95—menyeimbangkan kepentingan reward langsung dan jangka panjang. Agar stabil, nilai return biasanya dinormalisasi sehingga menghasilkan *advantage* dari setiap aksi, yang kemudian menjadi dasar metode REINFORCE. Algoritma semacam ini menjalankan banyak episode, menghitung advantage, lalu memperbarui parameter kebijakan berdasarkan efektivitas tindakan. Dengan cara ini, agen dapat mencapai durasi keseimbangan maksimum pada tugas CartPole.



Bab ini juga memperkenalkan kerangka matematis **Markov Decision Process (MDP)** yang menjadi fondasi RL. Pada MDP, keadaan berubah secara probabilistik tergantung kondisi saat ini dan aksi yang dipilih. Persamaan optimalitas Bellman (Persamaan 1) merumuskan nilai suatu keadaan secara rekursif. Dari sini muncul pendekatan *dynamic programming* seperti value iteration (Persamaan 2). Ketika nilai diturunkan ke tingkat tindakan, muncullah **Q-value**, yang menyatakan reward terharapkan dari suatu aksi pada keadaan tertentu. Dengan memperbarui Q-value secara iteratif, agen dapat menemukan kebijakan optimal. Gambar 4 menggambarkan bagaimana agen harus mempertimbangkan risiko dan potensi hasil saat menghadapi transisi yang tidak pasti.



Dalam situasi nyata, kita biasanya tidak mengetahui probabilitas transisi lingkungan. Karena itu, **Temporal Difference (TD) learning** digunakan untuk memperbarui nilai secara bertahap berdasarkan pengalaman langsung. **Q-learning** memperluas ide ini dengan memperbarui Q-value dari setiap pengamatan, bahkan saat agen masih melakukan eksplorasi acak. Strategi seperti *ε-greedy* memastikan agen tetap mencoba tindakan baru sembari memanfaatkan pengetahuan yang sudah diperoleh. Beberapa metode modern menambahkan bonus berbasis rasa ingin tahu (*curiosity-driven*

exploration) agar agen terdorong mengunjungi keadaan yang jarang dialami.

Untuk lingkungan yang besar atau berdimensi kontinu, tabel Q-value tidak lagi memadai, sehingga diperlukan **fungsi aproksimasi**. Pendekatan seperti **Deep Q-Networks (DQN)** yang diperkenalkan DeepMind pada tahun 2013 menjadi terobosan penting karena mampu belajar langsung dari data mentah seperti piksel.

Untuk mengatasi masalah tersebut, berbagai teknik stabilisasi dikembangkan. *Target networks* digunakan untuk menghasilkan Q-value acuan yang lebih stabil. **Double DQN** diperkenalkan untuk mengurangi bias estimasi berlebih dengan memisahkan proses pemilihan aksi dan evaluasi nilai. **Prioritized experience replay** mengutamakan pengalaman yang lebih informatif untuk dipelajari, sehingga mempercepat konvergensi. Berkat inovasi-inovasi tersebut, agen mampu mengungguli pemain manusia dalam banyak game Atari dan menjadi fondasi bagi pencapaian besar seperti kompetisi Go oleh AlphaGo.

Meskipun kemajuan reinforcement learning sangat signifikan, metode ini tetap memiliki tantangan besar: hasil yang tidak stabil, kebutuhan data yang tinggi, ketergantungan pada hyperparameter, dan sensitivitas terhadap desain reward. Pelatihan sering kali menghabiskan banyak waktu dan komputasi. Namun, terlepas dari kesulitannya, RL terus berkembang dan diterapkan pada berbagai sektor dunia nyata seperti optimasi pusat data Google, robotika, dan sistem rekomendasi. Dengan semakin matangnya metode dan meningkatnya kemampuan komputasi, reinforcement learning tetap menjadi salah satu jalur paling menjanjikan dalam pencapaian kecerdasan buatan yang mandiri dan adaptif.