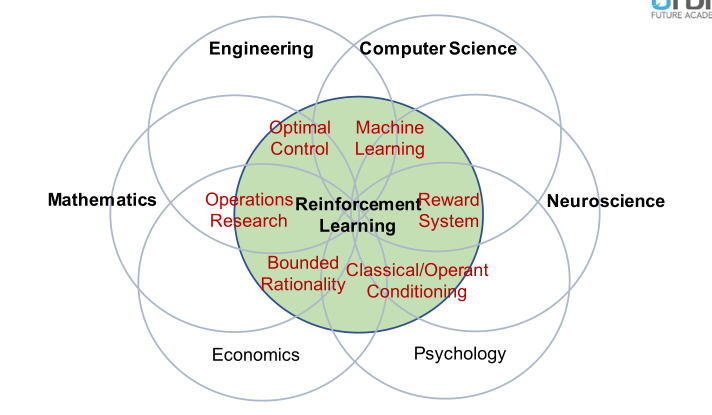
**Reinforcement Learning**

Nama : Ivan Febriansyah  
Kelas : Visioner

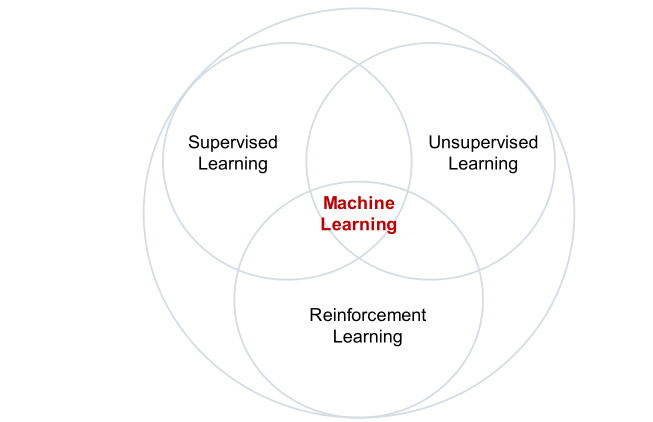
1. **Definisi dan sejarah Reinforcement learning**

Reinforcement learning adalah sub bidang machine learning yang mengajarkan agen cara memilih tindakan dari ruang tindakannya, dalam lingkungan tertentu, untuk memaksimalkan reward dari waktu ke waktu.

Reinforcement learning dalam berbagai bidang



Cabang Machine learning



Reinforcement Learning has four essential elements:

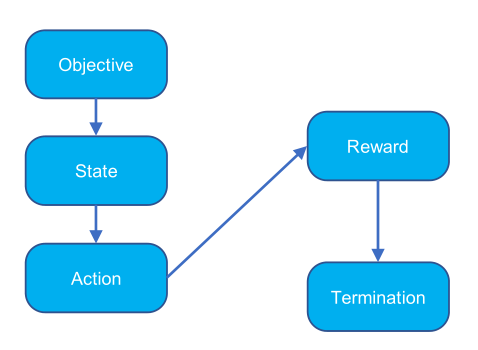
**Agent** : The program you train, with the aim of doing a job you specify.

**Environment** : The world, real or virtual, in which the agent performs actions.

**Action** : A move made by the agent, which causes a status change in the environment.

**Rewards** : The evaluation of an action, which can be positive or negative.

1. **Algoritme RL**

  
**Algoritme RL**

* Algoritme RL menghendaki pertukaran state-action-reward (st, at, rt) antara agent dan environment. Proses ini dinamakan sequential decision-making process, dan (st, at, rt) dinamakan experience.
* RL menghitung jumlah dari reward yang diterima oleh agent. Tujuan (objective) dari agent adalah memaksimalkan total reward.
* RL belajar (learn) dari interaksi agent dengan environment menggunakan proses trial & error dan reward yang diperoleh agent, untuk memperkuat (reinforce) aksi positif.
* Agent berinteraksi terhadap environment dengan melakukan sebuah aksi/tindakan dari satu kondisi ke kondisi yang lain.
* Agent akan menerima reward berdasarkan aksinya.
* Berdasarkan reward tersebut, agent akan memahami apakah aksinya baik atau buruk.
* Jika aksinya baik, agent akan menerima reward positif, sehingga agent tersebut akan cenderung lebih memilih melakukan aksi yang serupa dengan aksi tersebut (exploitation), atau melakukan aksi lain yang dapat menghasilkan reward positif.

Ilustrasi reward pada pergerakan Agent RL



1. **Element dan Environment RL**

Di luar dari Agent dan Environment yang merupakan sebagai element utama,

ada 4 sub elemen sebagai penyusun utama sistem Reinforcement Learning:

Elemen utama :

* 1. **Agent**, yang dimaksud dengan agent adalah perangkat atau *software* (perangkat lunak) yang dapat belajar dari *environment*.
  2. **Environment,** sedangkan yang dimaksud dengan *environment* adalah segala sesuatu yang ada di luar agent yang dimana sebagai tempat agent untuk melakukan *exploration* dan *exploitation*.

Sub Elemen utama :

1. **Policy**, rules (aturan) atau strategi yang digunakan oleh agent untuk melakukan action (**A**) selanjutnya, berdasarkan state (**S**) saat ini.
2. **Reward Signal**, *feedback* atau umpan balik untuk agent. *Reward* dapat bernilai positif (berupa hadiah) atau negatif (berupa hukuman) dan juga nol (tidak ada tindakan apapun terhadap *agent*).
3. **Value function (V) s**, total nilai jangka panjang yang diharapkan (*expected long-ter return without discount*) dari state saat ini di bawah *policy 𝜋*. Value ini adalah kebalikan dari *short-term reward* (R).

* Optimal Value Function adalah sebuah fungsi yang memiliki nilai tertinggi untuksemua state dibandingkan dengan fungsi nilai lainnya.
* Optimal Policy adalah policy (kebijakan) yang memiliki fungsi nilai optimal.

1. **Model environment**, segala sesuatu yang meniru perilaku environment. Atau, secara umumnya, sebuah kesimpulan tentang bagaimana perilaku dari environment.

Model Environment dibagi menjadi dua, yaitu :

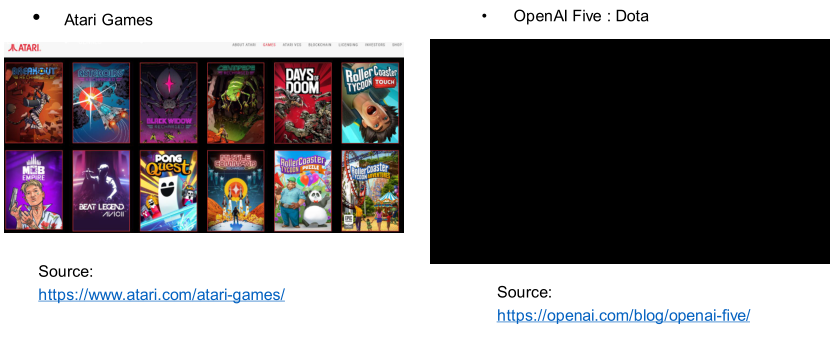
* Model Based RL
* Model Free RL

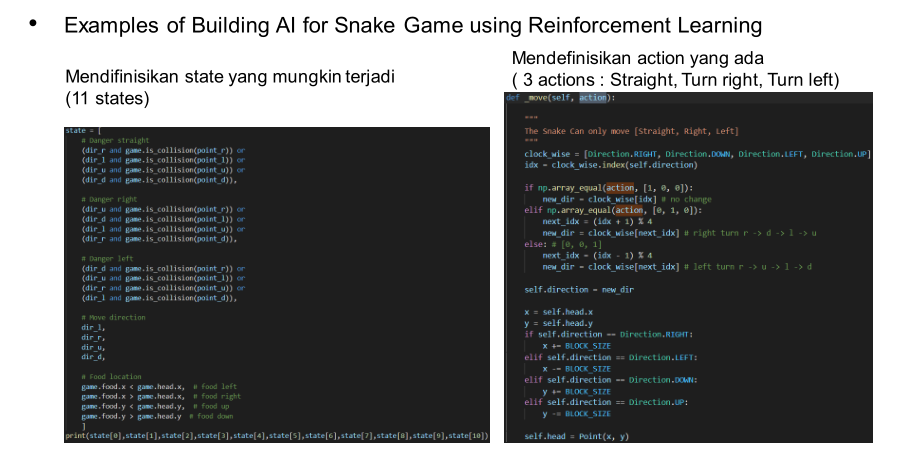
Dalam memecahkan persoalan yang berhubungan dengan Reinforcement Learning, ada dua metode yang dipakai, yaitu model base dan model free, keduanya didasarkan pada bagaimana lingkungannya yang dihadapi oleh agent.

* **Model Based RL**, model ini adalah model dasar atau paling sederhana dan memiliki perencanaan terhadap tindakannya, atau dapat juga dimaknai sebagai agent mengeksploitasi informasi yang dipelajari sebelumnya untuk menyelesaikan tugasnya.
* **Model Free RL**, sedangkan model ini adalah kebalikan dari model Model Based RL, karena agent belajar dari lingkungan melalui metode trial and error untuk memperoleh pengalamannya.

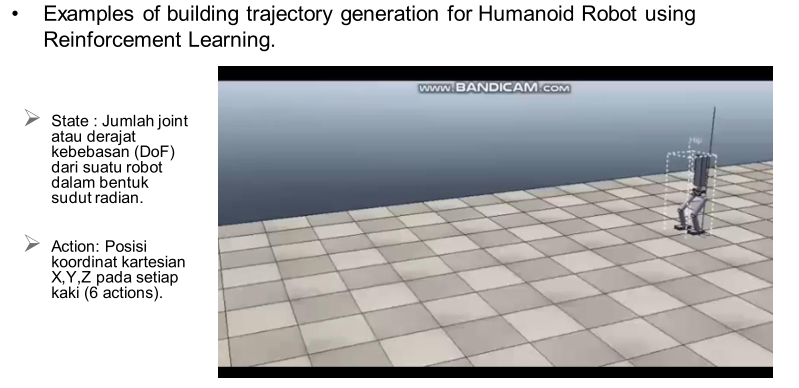
1. **Aplikasi RL**

Contoh Reinforcement learning untuk game :





Contoh reinforcement learning untuk robotics :

****

1. **Markov Decision Progress (MDP) dan dynamic programming.**

Markov Property

“The future is independent of the past given the present”

Sebuah keadaan 𝑆𝑡 bisa disebut Markov jika dan hanya jika:

Sebuah state memiliki informasi dari sejarah dan Ketika state diketahui, sejarah dapat diabaikan

Apa itu MDP ?

Markov Decision Process merupakan sebuah tuple (S, A, P, R, γ).

Dimana:

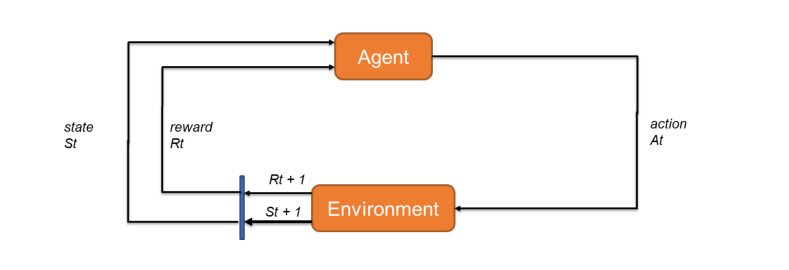
* **S** merupakan state
* **A** merupakan action
* **P** merupakan state transition probability function (transition probability)
* **R** merupakan reward function
* **γ** merupakan discount factor (γ ϵ [0, 1])

Markov Decision Processes mendeskripsikan secara formal lingkungan untuk RL

Secara spesifik biasanya dibuat saat environment fully observable

Hampir semua RL problems dapat diformalisasi menggunakan MDP

* Optimal Control
* Partially Observable Program
* Bandits Problem



1. **Monte Carlo Prediction**

Konsep Monte Carlo (MC)

* MC tidak mengambil pengetahuan lengkap dari environment.
* MC belajar dari experience, episode per episode, baik itu experience aktual, maupun simulasi.
* MC belajar dari episode-episode secara utuh dan independent, tidak bootstrapping.
* MC didefinisikan untuk jenis episodic environment.
* Ide utama dari MC: Value didapatkan dari rata-rata returns.
* Dengan semakin banyak returns, nilai rata-ratanya diharapkan konvergen pada expected value.
* Seluruh episode dipertimbangkan dalam MC.
* Hanya satu pilihan setiap perpindahan state di MC, sedangkan DP mempertimbangkan semua probabilitas transisi pada setiap perpindahan state.
* Estimasi-estimasi untuk semua state adalah independent di MC, tidak bootstrap.
* Waktu yang dibutuhkan untuk mengestimasi suatu state tidak bergantung pada jumlah total state.

Element Algoritme MC

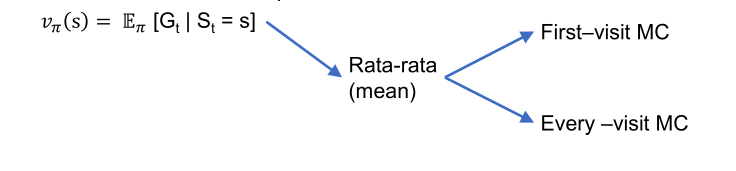
* **Goal** :: Belajar 𝑣𝜋 melalui episode-episode dariexperience yang patuh pada policy 𝜋 :

S1, A1, R2, …, St ~ 𝜋

* **Return** :: Adalah total discounted reward :

Gt = Rt+1 + 𝛾Rt+1 + … + 𝛾𝑇−1 RT

* **Value Function** :: Adalah expected return :



Perhitungan Value Function di MC :

* Total counter N(s) N(s) + 1
* Total return S(s) S(s) + Gt
* Value V(s) = S(s) / N(s)
* V(s) konvergen ke 𝑣𝜋 sejauh N(s) mendekati ∞