Nama: Irvan Naufali Rahmanto
Kelas : IF-40-08

1. **Analisa masalah**

   Dalam permasalahan ini kita ditujuakan untuk membangun sebuah program *Q-learning* untuk menemukan *optimum policy* dimana *Agent* yang berada di posisi start (1,1) mampu menmukan *goal* yang berada di posisi (15,15) dengan mendapat total *reward* maksimum pada *grid* yang disediakan. Pada kasus ini agent hanya bisa melakukan empat aksi: N,E,S dan W yang secara berurutan menyatakan *North* (ke atas) , *East* (ke kanan) , *south* (ke bawah) dan *west* (ke kiri). Kita diperbolehkan menggunakan skema apapun dalam mengimplementasikan sebuah episode ini.

2. **Strategi penyelesaian**

   1) **Mengatur *state* dan *actions***
      Pada step pertama penyelesaian masalah *Q-learning* kita terlebih dahulu mengetahui *state* dari pada permasalahan, disini terdapat *states* yang berada ditabel berukuran 15x15 = 225, kemudian selanjut nya kita tentukan *actions* untuk aturan berjalan nya agent pada state, kita tentukan 4 *actions* yaitu N (ke atas), S (ke bawah). E (ke kanan) dan W (ke kiri).

   2) **Inisialisasi Q(s,a) *function***
      Selanjut nya dalam memudahkan aturan setelah mengatur *state* dan *action* kita dapatkan bahwa dibutuhkan *function* untuk menjalankan kedua aturan sebelum nya yaitu *state* dan *action*, dari table yang sudah kita ketahui didapatkan bahwa insialisasi yang dilakukan sebagai algoritma berikut.

      Initialize $Q(s, a)$ arbitrarily
      Repeat (for each episode)
      Initialize s
      Repeat (for each step of the episode)
      Choose a from s using an exploratory policy
      Take action a, observe r, s'

   3) **Kalkulasi nilai Q(s,a)**
      Pada berjalan nya proses inisialisasi kita diharus kan dalam menghitung perpindahan inisialisasi pada table yang dilakukan dengan mengkalkulasi nilai baru / *new values* dari Q(s,a), dari sana kita menghitung *new values* dengan rumus berikut,

      $$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$
      $$s \leftarrow s'$$

   4) **Perbarui *function* Q(s,a)**
      Niali fungsi / *function* Q(s,a) yang sudah kita buat dan kita inisialisasi pada step sebelum nya kita akan terus *update* dengan rumus pada step sebelum nya, jadi diibaratkan tiap *agent* berjalan melewati kotak kotak pada tabel maka tabel tabel

sebelum nya akan diperarui sesuai berjalan nya *agent* yang menuju goal state, tentu nya menggunakan rumus kalkulasi pada step sebelum nya.

5) **Dapatkan *optimum policy***
Setelah berjalan nya *agent* dari *start state* menuju *goal state* ,maka kita bisa dapatkan berjalan nya agent dari awal menuju tujuan nya sebagai *optimal policy*, dimana proses berjalan *agent* tersebut menuju *final state* sesuai *state* dan *action* yang sudah kita tentukan, maka disimpulkan bahwa *state* dan *action agent* menuju *goal state* sebagai *optimum policy* dari episode *problem* ini, berdasarkan step step sebelum nya.

3. Hasil percobaan

- **Outputan hasil :**

```
goal [[14, 0], [14, 1], [13, 1], [12, 1], [12, 2], [11, 2], [10, 2], [10, 3], [9, 3], [8, 3], [8, 4], [8, 5], [8, 6], [7, 6], [6, 6], [6, 7
], [5, 7], [5, 8], [5, 9], [5, 10], [4, 10], [3, 10], [3, 11], [2, 11], [2, 12], [1, 12], [1, 13], [1, 14], [0, 14]]
```

- **Maksimum policy :**

```
reward 452
```