

Problem :

Bangunlah sebuah program Q-learning untuk menemukan optimum policy sehingga Agent yang berada di posisi Start (1,1) mampu menemukan Goal yang berada di posisi (15,15) dengan mendapatkan Total Reward maksimum pada grid world Gambar di bawah ini. 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)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| 0 | -1 | -2 | -3 | -2 | -3 | -3 | -4 | -1 | -4 | -2 | -1 | -2 | -3 | -3 | 500 |
| 1 | -1 | -3 | -1 | -2 | -4 | -1 | -4 | -1 | -4 | -2 | -4 | -2 | -2 | -2 | -1 |
| 2 | -4 | -2 | -1 | -4 | -2 | -1 | -2 | -4 | -2 | -3 | -2 | -1 | -2 | -4 | -4 |
| 3 | -4 | -2 | -4 | -1 | -3 | -2 | -3 | -2 | -4 | -2 | -4 | -1 | -2 | -4 | -2 |
| 4 | -4 | -2 | -2 | -3 | -2 | -3 | -1 | -1 | -4 | -2 | -1 | -3 | -4 | -2 | -4 |
| 5 | -4 | -3 | -3 | -4 | -2 | -3 | -4 | -2 | -2 | -1 | -1 | -2 | -1 | -2 | -1 |
| 6 | -2 | -3 | -2 | -1 | -1 | -3 | -2 | -1 | -4 | -3 | -1 | -1 | -2 | -3 | -3 |
| 7 | -3 | -1 | -1 | -4 | -4 | -3 | -1 | -2 | -3 | -1 | -1 | -4 | -4 | -3 | -3 |
| 8 | -3 | -1 | -4 | -2 | -3 | -3 | -1 | -4 | -4 | -4 | -2 | -2 | -2 | -2 | -1 |
| 9 | -3 | -4 | -4 | -2 | -3 | -4 | -3 | -3 | -2 | -2 | -3 | -4 | -3 | -4 | -1 |
| 10 | -3 | -4 | -1 | -1 | -1 | -4 | -4 | -4 | -4 | -1 | -2 | -4 | -2 | -2 | -1 |
| 11 | -1 | -3 | -3 | -3 | -3 | -3 | -3 | -3 | -4 | -1 | -2 | -4 | -1 | -2 | -4 |
| 12 | -2 | -2 | -1 | -2 | -2 | -2 | -4 | -3 | -1 | -4 | -1 | -4 | -2 | -2 | -2 |
| 13 | -2 | -1 | -3 | -1 | -4 | -4 | -1 | -3 | -3 | -1 | -1 | -2 | -3 | -4 | -3 |
| 14 | -2 | -2 | -1 | -4 | -4 | -4 | -2 | -2 | -3 | -1 | -2 | -2 | -1 | -1 | -3 |

Strategi Penyelesaian Masalah:

Strategi yang dilakukan yaitu dengan mengimplementasikan metode **Q-Learning** kedalam sebuah program sehingga program tersebut dapat mencapai titik 15,15 dengan mendapatkan Total Reward maksimum. Dalam program yang saya buat , saya membuat tabel Q(s,a) terlebih dahulu untuk mengisitung nilai N,S,E,dan Wnya, setelah sudah di buat tabel Q(s,a) saya menentukan titik tersebut dapat berjalan ke arah mana saja dan jika sebelumnya terdapat arah yang berjalan ke dia maka arah sebaliknya akan di hapus untuk mengatasi terjadinya looping pada perjalanan. Penentuan jalan kemananya akan di random arah jalannya , jika sudah terpilih arah mana yang akan di lewati maka arah tersebut akan mencari nilai Q(s,a) nya dengan rumus

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

setelah sudah mendapatkan hasil dan di simpan di tabel Q(s,a) maka titik akan di upate sesuai arahnya. Saya menggunakan 500 episode dengan 200 state jalan, setelah sudah selesai mengupdate tabel Q maka selanjutnya saya memilih jalan sesuai max pada tabel Q(s,a) tetapi jika titik tersebut sudah pernah di lewati maka titik tersebut tidak akan dapat melewati titik tersebut kembali , cara ini mengatasi untuk tidak terjadi looping dan dapat mencapai titik goal.

Analisis :

Berdasarkan penjelasan yang saya jabarkan maka dapat diketahui bahwa pemilihan nilai gamma dan alfa berperan penting untuk mendapatkan nilai yang optimum serta pemilihan berapa episode dan state juga berperan penting untuk dapat menjelajahi setiap titik. Tidak kalah derperan penting yaitu pengecekan titik yang sudah di jelajahi , pengecekan ini sangat penting karna dapat mencegah looping yang terjadi sehingga tidak dapat mencapai titik goal.

Hasil Percobaan :

Berikut beberapa hasil percobaan yang dilakukan :

```
print('Titik Akhir ',x,y)
print('Total Reward',total)
```

```
('Titik Akhir ', 14, 0)
('Total Reward', 452)
```

```
In [395]: t = pd.DataFrame(total2).T
header = t.iloc[0]
t = t[1:]
t = t.rename(columns = header)
t
```

```
Out[395]:
```

| | 210 | 195 | 180 | 165 | 150 | 135 | 120 | 121 | 106 | 107 | 92 | 93 | 94 | 95 | 96 | 97 | 82 | 83 | 84 | 85 | 70 | 71 | 56 | 41 | 26 | 27 | 28 | 29 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | -2 | -2 | -2 | -1 | -3 | -3 | -3 | -1 | -1 | -1 | -2 | -1 | -1 | -3 | -2 | -1 | -2 | -2 | -1 | -1 | -1 | -3 | -1 | -1 | -2 | -2 | -2 | -1 |

```
In [394]: df = pd.DataFrame(perjalanan).T

header = df.iloc[0]
df = df[1:]
df = df.rename(columns = header)
df
```

```
Out[394]:
```

| | 210 | 195 | 180 | 165 | 150 | 135 | 120 | 121 | 106 | 107 | 92 | 93 | 94 | 95 | 96 | 97 | 82 | 83 | 84 | 85 | 70 | 71 | 56 | 41 | 26 | 27 | 28 | 29 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | N | N | N | N | N | N | E | N | E | N | E | E | E | E | N | E | E | E | N | E | N | N | N | E | E | E | N | |
| 2 | 195 | 180 | 165 | 150 | 135 | 120 | 121 | 106 | 107 | 92 | 93 | 94 | 95 | 96 | 97 | 82 | 83 | 84 | 85 | 70 | 71 | 56 | 41 | 26 | 27 | 28 | 29 | 14 |

```
print('Titik Akhir ',x,y)
print('Total Reward',total)
```

```
('Titik Akhir ', 14, 0)
('Total Reward', 452)
```

```
In [402]: t = pd.DataFrame(total2).T
header = t.iloc[0]
t = t[1:]
t = t.rename(columns = header)
t
```

```
Out[402]:
```

| | 210 | 211 | 212 | 197 | 182 | 167 | 152 | 153 | 154 | 139 | 124 | 125 | 126 | 111 | 96 | 97 | 82 | 83 | 84 | 85 | 70 | 71 | 56 | 41 | 26 | 27 | 28 | 29 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | -2 | -2 | -1 | -3 | -1 | -3 | -1 | -1 | -1 | -3 | -3 | -3 | -1 | -1 | -2 | -1 | -2 | -2 | -1 | -1 | -1 | -3 | -1 | -1 | -2 | -2 | -2 | -1 |

```
In [403]: df = pd.DataFrame(perjalanan).T

header = df.iloc[0]
df = df[1:]
df = df.rename(columns = header)
df
```

```
Out[403]:
```

| | 210 | 211 | 212 | 197 | 182 | 167 | 152 | 153 | 154 | 139 | 124 | 125 | 126 | 111 | 96 | 97 | 82 | 83 | 84 | 85 | 70 | 71 | 56 | 41 | 26 | 27 | 28 | 29 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | E | E | N | N | N | N | E | E | N | N | E | E | N | N | E | N | E | E | E | N | E | N | N | N | E | E | E | N |
| 2 | 211 | 212 | 197 | 182 | 167 | 152 | 153 | 154 | 139 | 124 | 125 | 126 | 111 | 96 | 97 | 82 | 83 | 84 | 85 | 70 | 71 | 56 | 41 | 26 | 27 | 28 | 29 | 14 |

