

# Laporan Analisis Sistem Pencari Jalur Optimal berdasarkan Optimum Policy menggunakan Metode Q-Learning.

Fauzan Firdaus – 1301164317 – IF-40-04

## A. Abstrak

Diketahui sebuah dataset berupa blok dengan ukuran baris sebanyak 15 dan kolom sebanyak 15. Misalkan terdapat objek/agen A yang berada di titik start. Titik start sebuah objek berada di pojok kiri bawah blok tersebut, sedangkan titik finish atau tujuan keberadaan objek/agen adalah di blok pojok kanan atas. Dengan menggunakan metode Q-Learning, program dapat mencari jalur optimal yang didapatkan dari *optimum policy* dari hasil perbaruan Q-Table pada beberapa episode jalur berbeda yang telah dilakukan. Bahasa yang digunakan untuk program ini adalah bahasa **Matlab**, dengan versi tools adalah **R2018A**.

## B. Q-Learning

Sedikit bahasan mengenai Q-Learning, Q-Learning adalah salah satu metode / algoritma dalam reinforcement Learning yang bertujuan untuk membuat *optimum policy* berupa aturan sebagai tindakan apa yang harus dilakukan oleh sebuah objek/agen untuk mencari solusi dalam sebuah tujuan tertentu.

Dalam program ini, output dari Q-Learning adalah mencari jarak yang paling optimal dan reward yang didapatkan berdasarkan jalur yang telah dilalui.

## C. Strategi Penyelesaian Masalah

Dalam penyelesaian permasalahan kali ini, Q-Learning digunakan untuk mendapatkan sebuah kebijakan/aturan objek/agen untuk berpindah pada suatu blok ke blok lainnya yang tersimpan di QTable. Maksudnya, untuk setiap blok yang ada pada sebuah grid dapat ditentukan arah dari objek/agen tersebut bergerak.

Pada kasus permasalahan kali ini terdapat 4 aksi yang dapat dilakukan, yaitu bergerak ke atas, bawah, kiri dan kanan. Namun ada beberapa kondisi yang menyebabkan objek/agen hanya bergerak 2 atau 3 aksi saja. Objek/agen hanya bisa melakukan 2 aksi saja ketika keberadaannya ada di pojok kiri bawah, pojok kiri atas, dan pojok kanan bawah. Sedangkan objek/agen bisa melakukan 3 aksi ketika berada di setiap sudut (bukan pojok) atas, bawah, kiri dan kanan.

Selain menentukan aksi yang akan dilakukan objek/agen, hal lain yang harus diperhatikan adalah Q-Table. Seperti yang sudah dijelaskan sebelumnya, Q-Table diperlukan sebagai *optimum policy*. Ketika objek/agen bergerak, maka isi dari Q-Table akan diperbarui sesuai baris arah pada Q-Table dan blok awal sebelum objek/agen tersebut pindah.

Dalam program ini, terdapat jumlah episode sebanyak 300. Dalam tiap episode, objek/agen akan bergerak dari titik start/awal sampai titik finish/tujuan. Setelah semua episode tersebut berjalan, Q-Table akan menjadi *optimum policy*.

Dari hasil akhir update Q-Table tersebut akan didapatkan jalur optimal untuk permasalahan kali ini. Dengan Q-Table sebagai *optimum policy*, setiap blok dapat menentukan arah objek/agen tersebut bergerak. Setiap blok memiliki 4 kemungkinan asli, maka arah yang harus dilakukan oleh objek/agen tersebut bergerak harus memilih nilai yang terbesar dari blok yang bersangkutan pada Q-Table.

Setelah mendapatkan arah yang harus dilakukan pada setiap blok, maka yang harus dilakukan selanjutnya adalah membuat jalur atau episode terbaik yang akan dilakukan. Episode terbaik yang akan dilakukan tersebut didapatkan dari arah setiap blok yang telah ditentukan secara iteratif. Misalkan, objek/agen di blok A terdefinisikan untuk bergerak ke atas menuju blok B, lalu blok B harus bergerak ke kanan menuju blok C, hingga seterusnya sampai titik finish dicapai.

Jalur optimal/episode terbaik kini sudah kita dapatkan, maka untuk menunjukkan jalur tersebut sudah baik, maka dihitunglah reward pada jalur tersebut. Reward didapatkan dari perhitungan/akumulasi setiap objek/agen tersebut singgah pada suatu blok.

#### D. Hasil atau Output Program

Penulis melakukan eksekusi program yang keluaran dari program tersebut berupa reward dan gambaran jalur objek tersebut bergerak. Berikut adalah outputnya.

Jalur objek bergerak berdasarkan optimum policy:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reward yang didapatkan :

453