

1. Jelaskan secara umum algoritma minimax !

Algoritma minimax merupakan algoritma searching (adversarial) untuk mencari langkah optimal bagi seorang pemain dengan asumsi bahwa lawan juga mengambil langkah optimal.

Algoritma minimax bekerja mirip seperti algoritma searching klasik. Bedanya, jika algoritma searching klasik memilih simpul anak dengan strategi yang konstan (misalnya memilih yang maksimum secara terus-menerus), algoritma minimax bekerja dengan strategi yang bergantian. Misalkan pada level 0 (akar) algoritma mencari langkah optimum, maka pada level 1, algoritma akan mencari langkah minimum.

2. Jelaskan bagaimana algoritma minimax mengambil langkah terbaik dalam permainan TicTacToe yang dibuat!

Pada permainan TicTacToe yang dibuat, agen bekerja sebagai player *bot* yang melawan player manusia. Skor sebuah simpul akan dihitung berdasarkan menang atau kalah player pada akar. Selain itu, nilai maksimum akan diberikan kepada simpul daun yang memenangkan player akar dengan jarak ke akar yang tidak berbeda jauh (semakin cepat menang semakin tinggi skornya).

Kedalaman atau “depth” pada kode adalah sebuah parameter yang menunjukkan maksimal kedalaman yang bisa dicapai dengan kondisi sekarang. Artinya, jika hanya tersisa 2 tempat kosong, kedalamannya yang dapat dicapai adalah 2. Kedalaman ini juga akan digunakan untuk menghitung skor. Semakin kecil parameter *depth*, berarti semakin dekat letaknya dengan akar. Ketika agen kalah, diberi nilai -1, seri 0, dan menang $1 + depth$.

Setelah mengetahui cara menghitung skor, maka algoritma minimax dapat dijalankan seperti pada pertanyaan pertama (selang-seling: min, max, min, max sampai ke daun). Bot akan mengambil nilai optimum dengan asumsi bahwa kita (player) melakukan gerakan yang optimum juga. *State space tree* akan di-*expand* sampai daun, tetapi kita dapat melakukan pruning menggunakan *alpha-beta* pruning. Sebagai ilustrasi, alpha-beta pruning berguna dalam kondisi berikut.

