

LAPORAN

PROYEK KECERDASAN BUATAN



Oleh

VINCENTIUS ALFREDO - C14180159

NICHOLAS BILLY - 26416038

GREGORIUS HANS - C14180051

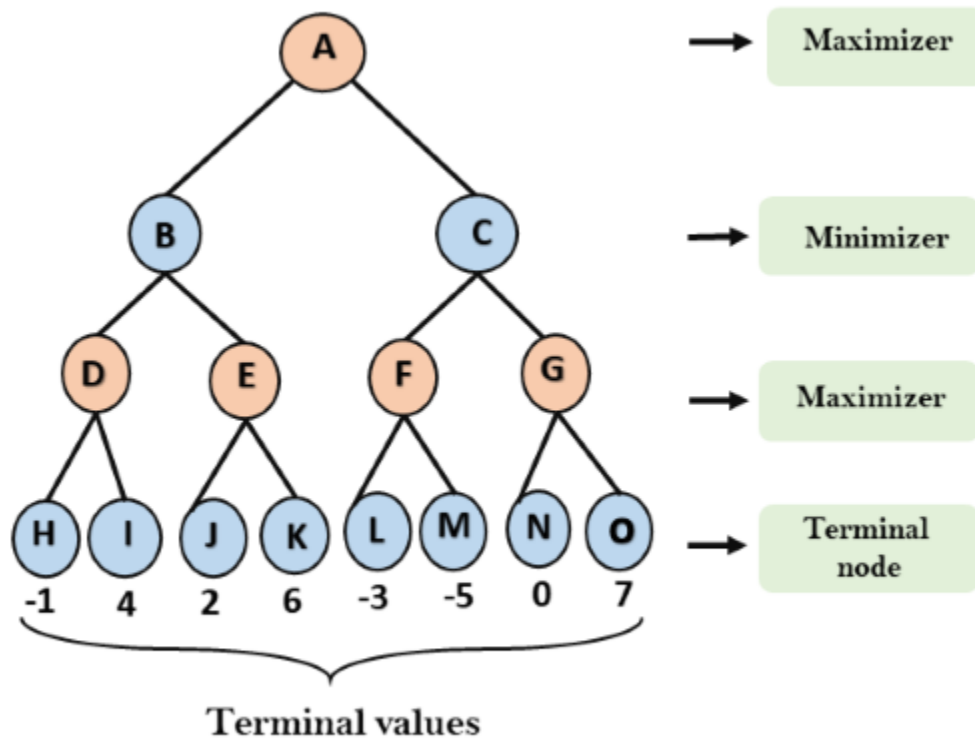
UNIVERSITAS KRISTEN PETRA

SURABAYA

2021

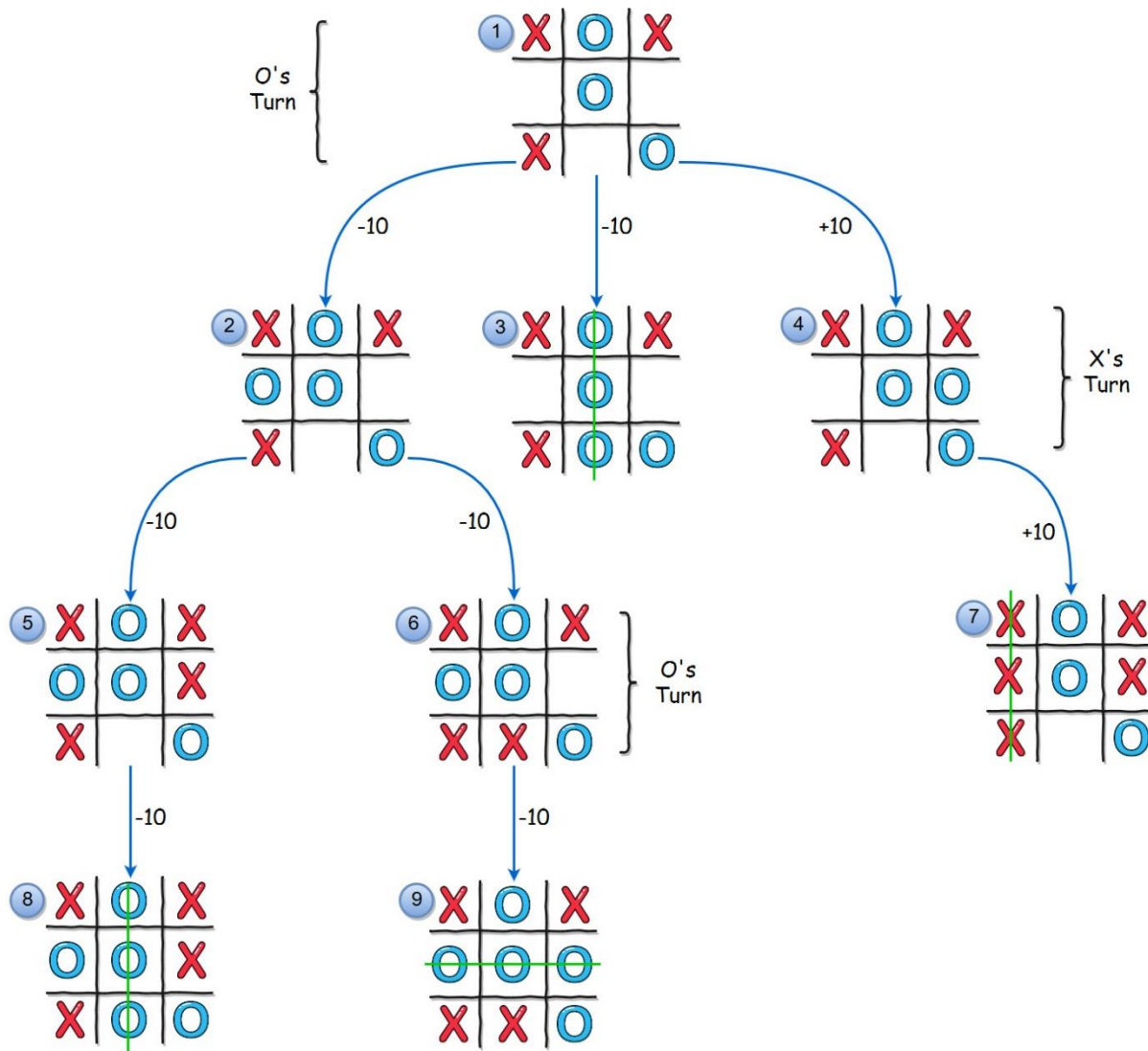
ALGORITMA MINIMAX

Algoritma Minimax dapat digunakan untuk menentukan gerakan / keputusan terbaik yang dapat diambil oleh sebuah AI yang dapat di implementasikan kedalam sebuah game yang dimainkan oleh 2 player (TicTacToe / Othello). Algoritma ini sendiri dapat dijelaskan penggunaanya seperti berikut :



Dari gambar diatas dapat dijelaskan bahwa pada node terminal (H dan seterusnya) merupakan node yang memiliki nilai sedangkan node atasnya (D dan node atasnya) tidak memiliki nilai, ini karena algoritma minimax mengambil sebuah nilai node yang sudah ditentukan oleh pengerjaannya, seperti pada gambar pada langkah pertama (node D) yang bergerak adalah maximize nya, oleh karena itu dari 2 kemungkinan yang dimiliki oleh node D yaitu node H dan node I, yang diambil adalah nilai yang maximal dari kedua node tersebut yaitu node I yang bernilai 4. Sedangkan pada node B, yang diambil adalah nilai minimize, karena algoritma ini akan selalu bergantian pengerjaannya, karena node B mencari nilai minimize maka dari 2 node yang ia miliki (node D bernilai 4 dan node E bernilai 6) maka nilai yang dimiliki oleh node B adalah 4.

IMPLEMENTASI TICTACTOE



Dari contoh diatas dapat dilihat bahwa algoritma minimax sangatlah efektif dalam implementasi game berbasis 2 player, dimana sebuah AI akan selalu mengambil keputusan terbaik berdasarkan informasi yang didapat. Gambar diatas dapat dijelaskan sebagai berikut : pada root (move 1) adalah gerakan AI / O dimana O adalah minimizer yang berarti ia akan mengambil move dengan nilai -10, dan Player / X adalah maximizer yang berarti ia akan mengambil move dengan nilai +10.

Dari gambar ini dapat dilihat bahwa O memiliki 3 gerak yang dapat ia ambil berdasarkan tempat kosong yang ada, dimana tiap gerakan ini akan ditandai dengan move 2, 3, dan 4. Pada move 3 AI menemukan bahwa ia dapat memenangkan pertandingan maka otomatis move 3 akan diberikan nilai -10 agar dapat diambil oleh sang AI. Tetapi jika AI mengambil move 2 dan 4 akan bercabang karena move 2 dan 4 tidak dapat langsung memenangkan pertandingan sehingga akan membuat cabang baru yang disebabkan oleh gerakan dari player. Karena move 2 dan 4 belum menang maka pada node ini tidak akan diberikan nilai, tetapi nilai yang mereka punya akan dilihat dari move selanjutnya.

Pada move 2 dapat dilihat bahwa move selanjutnya memiliki 2 kemungkinan lagi, yaitu move 5 dan 6. Pada tingkatan node ini akan dilakukan pengecekan giliran lagi untuk menentukan langkah yang diambil oleh AI, pada tingkatan ini adalah giliran O / AI dimana ia akan mengambil nilai -10 lagi. Tetapi pada move 4, AI dan player sama – sama memiliki kesempatan menang, tetapi karena gerakan pada tingkatan selanjutnya adalah gerakan player maka nilai yang direturn adalah nilai +10, karena meskipun pada move itu AI dapat menang, nyatanya pada saat move 7 tersebut adalah giliran player, maka otomatis AI tidak akan dapat menang pada move tersebut. Karena itu move 7 akan diberikan nilai +10 karena sudah terbukti bahwa player yang menang, karena move 7 adalah anak dari move 4 nilai dari move 7 akan direturn kan lagi ke move 4.