

IMPLEMENTASI ALGORITMA RUNUT BALIK PADA PERMAINAN TIC TAC TOE

Lola

*Program Studi Teknik Informatika, FTI, Institut Teknologi Budi Utomo Jakarta,
lola.rezak@gmail.com*

Abstrak

Permainan TicTacToe merupakan sebuah permainan papan untuk dua pemain yang dimainkan secara bergiliran. Pemain yang mendapat giliran mengisikan X atau O sesuai bagiannya pada kotak yang kosong. Pemain yang berhasil menempatkan 3 mark dalam garis horizontal, vertical, atau diagonal menjadi pemenangnya.

Kata kunci : TicTacToe, Minimax, Alpha-Beta Pruning, Algoritma Runut Balik.

1. PENDAHULUAN

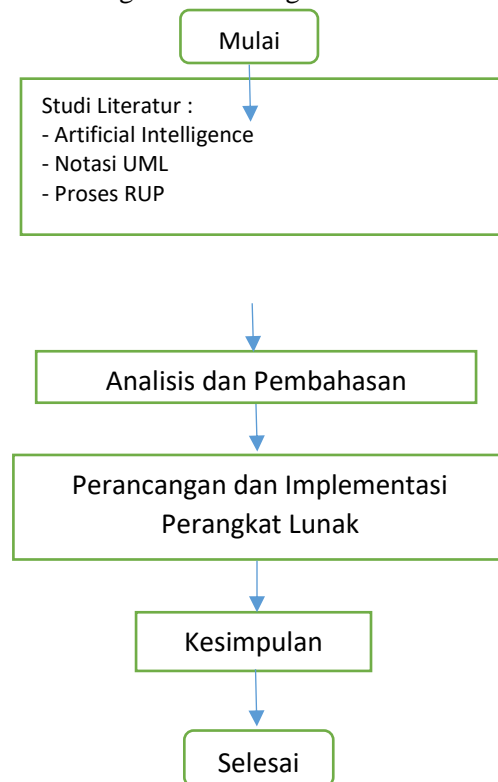
Tic Tac Toe, yang di Indonesia dikenal dengan istilah "*Catur Jawa*", adalah permainan yang bisa diselesaikan. Ini berarti ada strategi yang terbukti secara matematis mampu memenangkan permainannya. Dalam Catur Jawa, dua pemain yang mengikuti strategi yang benar akan selalu seri. Namun, Anda masih bisa menang terhadap pemain yang tidak menguasai strategi ini, terutama ketika mereka membuat kesalahan.

Pada permainan TicTacToe ini, ada pemain yang kalah, menang, atau keduanya berakhir imbang. Namun, bagaimana caranya agar kita selalu memenangkan pertandingan. Tentunya dengan bantuan komputer, kita akan sulit untuk dikalahkan dalam permainan ini.

Pada makalah ini akan dibahas program dan algoritma untuk mencari gerakan terbaik agar bisa memenangkan permainan.

2. METODOLOGI

Metodologi penelitian digambarkan dalam bentuk diagram alir sebagai berikut :



Gambar 1. Diagram Alir Metodologi Penelitian

Sumber :

https://www.researchgate.net/publication/33823569_5_Metode-Metode_Penelitian_Dalam_Penulisan_Jurnal_Ilmuah_Elektronik

3. HASIL DAN PEMBAHASAN

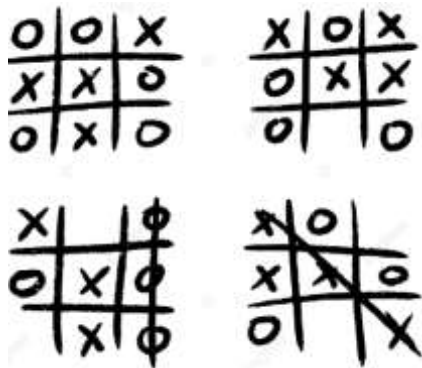
3.1 HASIL

Untuk dapat memahami konsep algoritma pemilihan langkah terbaik pada permainan TicTacToe dengan menggunakan algoritma backtracking, diperlukan beberapa pemahaman mengenai bidang berikut:

TicTacToe merupakan sebuah permainan papan yang dimainkan oleh dua orang pada sebuah papan kotak yang terdiri dari 9 (3x3) kotak. Permainan ini sangat sederhana, pemain yang pertama kali menggambarkan 3 O atau X pada satu garis lurus, menjadi pemenangnya.

Permainan ini hanya memerlukan sebuah kertas dan pensil, saat ini tic tac toe dapat dimainkan dengan mudah melalui perangkat seluler ataupun komputer.

TicTacToe 3x3 merupakan variasi klasik dari Tic Tac Toe. Permainannya terdiri dari 3 baris, 4 kolom, dan memerlukan 3 mark dalam satu garis sebagai syarat menang. Apabila seluruh kotak sudah terisi, dan tidak ada yang berhasil menempatkan 3 X atau 3 O dalam satu garis, maka permainan berakhir imbang.



Gambar 2. Permainan TicTacToe 3x3

Sumber:https://id.pngtree.com/freepng/hand-drawn-tic-tac-toe-game_8017217.html

Dalam sebuah permainan Tic Tac Toe yang terdiri dari 9 kotak, terdapat 9! kemungkinan posisi (urutan gerakan berpengaruh) yaitu sebesar 362880. Tentunya solusi dari permainan ini bisa dipecahkan dengan mudah dengan cara brute force. Telah dibuktikan menggunakan algoritma brute force, permainan ini bila dimainkan secara sempurna, akan selalu

memberikan hasil imbang, tidak peduli pemain, bermain sebagai X atau O (pertamasisi melakukan gerakan yang tidak akurat).

Algoritma runut balik (*backtracking*) dapat dipandang dalam dua hal. Yang pertama, algoritma backtracking merupakan sebuah fase di dalam algoritma traversal DFS. Algoritma backtracking juga dapat dipandang sebagai sebuah metode pemecahan masalah yang mangkus, terstruktur, sistematis, baik untuk persoalan optimasi maupun non optimasi. Pada makalah ini, akan digunakan definisi kedua algoritma backtracking, yaitu sebagai metode. Algoritma backtracking pertama kali diperkenalkan oleh D. H. Lehmer tahun 1950. Aturan umum Algoritma Backtracking:

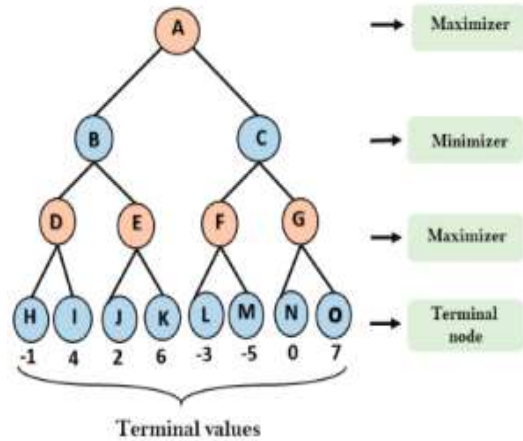
1. Solusi persoalan Solusi dinyatakan sebagai vector dengan n-tuple: $X = (x_1, x_2, \dots, x_n)$, $x_i \in S_i$. Umumnya $S_1 = S_2 = \dots = S_n$
2. Fungsi pembangkit nilai x_k
 - Dinyatakan sebagai predikat $T()$.
 - $T(x[1], x[2], \dots, x[k - 1])$ membangkitkan nilai untuk x_k , yang merupakan komponen vektor solusi.
3. Fungsi pembatas
 - Dinyatakan sebagai predikat $B(x_1, x_2, \dots, x_k)$.
 - B bernilai true jika (x_1, x_2, \dots, x_k) mengarah ke solusi. Mengarah ke solusi artinya tidak melanggar kendala (constraints).
 - Jika nilai true, maka pembangkitan untuk nilai x_{k+1} dilanjutkan, tetapi jika false, maka (x_1, x_2, \dots, x_k) dibuang.

3.2 PEMBAHASAN

Algoritma minimax merupakan sebuah algoritma backtracking yang digunakan untuk pengambilan sebuah keputusan sebuah permainan giliran, misalnya TicTacToe, catur, checkers, dll. Algoritma ini menyajikan Gerakan yang optimal untuk pemain, dengan asumsi lawan juga melakukan Gerakan yang optimal.

Dalam algoritma ini terdapat dua peran, yaitu maximizer yang selalu memilih cost maksimum, serta minimizer yang selalu memilih cost minimum. Algoritma ini

menerapkan *backtracking* dengan urutan pencarian *depth-first-search*.



Gambar 3. Algoritma Minimax
 Sumber : mini-max-algorithm-in-ai-step1.png (500×447) (jvatpoint.com)

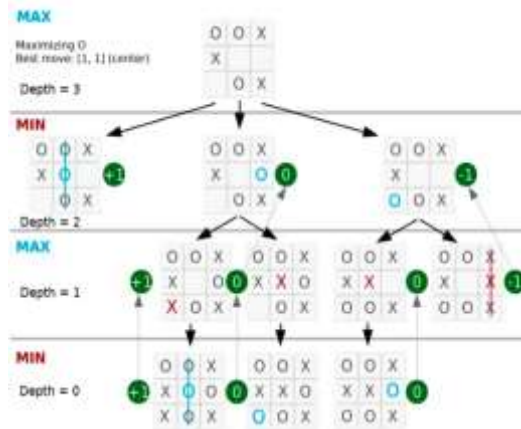
Terminal node atau daun merupakan state akhir sebuah permainan atau kedalaman maksimum yang dicapai.

Untuk menghitung cost dari node A, digunakan langkah sebagai berikut:

- Buat pohon keputusan untuk setiap kemungkinan gerakan hingga mencapai state akhir yang menghasilkan terminal node
- Hitung cost tiap terminal node
- Node D, E, F, G Maximizer: $D = \max(-1, 4) = 4$ $E = \max(2, 6) = 6$ $F = \max(-3, -5) = -3$ $G = \max(0, 7) = 7$
- Node B, C Minimizer: $B = \min(4, 6) = 4$ $C = \min(-3, 7) = -3$
- Node A Maximizer: $A = \max(4, -3) = 4$

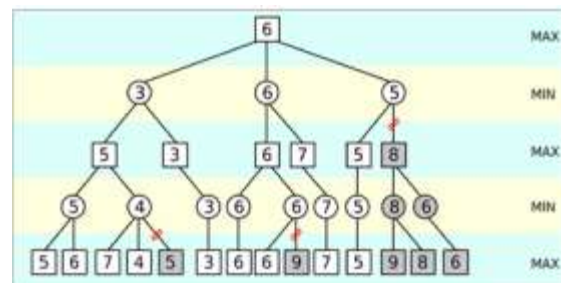
Dari hasil perhitungan, didapat gerakan yang optimal adalah ke kiri (ke Node B).

Minimax pada TicTacToe memiliki prinsip yang sama dengan Minimax pada umumnya. Pemain X, atau pemain yang jalan pertama berperan sebagai maximizer, sementara pemain O berperan sebagai minimizer. Mula-mula pohon keputusan dibuat hingga mencapai terminal node.



Gambar 4. Algoritma Minimax Tic Tac Toe
 Sumber: A. Nugraha
<https://viandwi24.medium.com/membuat-ai-tictactoe-dengan-algoritma-minimax-javascript-part-2-dan-mengenal-apa-itu-minimax-908c7f0a9f8c>

Alpha-Beta Pruning merupakan sebuah metode atau fungsi pembatas, untuk mengoptimasi algoritma backtracking (minimax). Dengan cara ini, program membatasi dan “membunuh” node yang dianggap tidak menuju ke solusi, dan menyisakan node yang berpotensi menuju solusi. Alpha-Beta Pruning dipastikan memberikan hasil yang sama dengan algoritma minimax biasa. Dengan menggunakan metode ini, sebuah program dapat melakukan pencarian solusi jauh lebih cepat dibanding tanpa menggunakan Alpha-Beta Pruning. Dengan demikian, program dapat melakukan pencarian untuk kedalaman yang lebih jauh lagi.



Gambar 5. Metode Alpha-Beta Pruning
 Sumber : Alpha-Beta-Pruning-Artificial-Intelligence.png (400×203) (thelinuxos.com)

Untuk setiap anak dari sebuah node. Apabila ditemukan node dengan cost yang lebih baik, maka node tersebut akan langsung “dibunuh”, hingga menyisakan node anak terbaik. Baik disini berarti lebih besar pada fase Maximizer atau lebih kecil pada fase Minimizer.

4. KESIMPULAN

Implementasi algoritma runut balik pada permainan TicTacToe memberikan pilihan/solusi terbaik secara lokal .dengan kedalaman tertentu. Solusi yang dihasilkan tidak selalu memberikan solusi optimum global karena keterbatasan komputer, sehingga pencarian perlu dibatasi hingga kedalaman tertentu.

Penerapan algoritma Minimax sudah sesuai untuk digunakan dalam pengambilan keputusan oleh *Artificial Intelligence* dalam permainan Tic Tac Toe.

Dengan menerapkan algoritma Minimax dalam permainan Tic Tac Toe, pemain pertama (manusia) tidak akan mendapatkan kemenangan ketika bermain melawan *Artificial Intelligence* tersebut dikarenakan *Artificial Intelligence* tersebut akan terus mencari langkah yang terbaik untuk menang.

DAFTAR PUSTAKA

- A. Nugraha, (2020). "Membuat AI TicTacToe dengan Algoritma Alpha-beta pruning — Javascript", Medium.
<https://viandwi24.medium.com/membuat-ai-tictactoe-dengan-algoritma-minimax-javascript-part-2-dan-mengenal-apa-itu-minimax-908c7f0a9f8c>
- "Artificial Intelligence | Alpha-Beta Pruning - Javatpoint",
<https://www.javatpoint.com/ai-alpha-beta-pruning>.
- "Artificial Intelligence | Mini-Max Algorithm - Javatpoint",
<https://www.javatpoint.com/mini-max-algorithm-in-ai>.