

Perbandingan Performa Algoritma *Minimax* Dan *Alpha-Beta Pruning* Pada *Game* Catur Cina

Marthin Hokuta Kurniawan¹, Daniel Udjulawa²

STMIK GI MDP; Jl. Rajawali No. 14 Palembang, Telp. (0711)376400 Fax/(0711)376360

Jurusan Sistem Informasi, STMIK GI MDP, Palembang

e-mail: *¹marthinhokitakurniawan@gmail.com, ²daniel@mdp.ac.id

Abstrak

Algoritma merupakan urutan yang lengkap dan logis, dengan urutan yang logis banyak cara yang dilakukan dengan urutan yang berbeda. Pada kasus ini akan dibandingkan performa dari algoritma Minimax dan Alpha Beta Pruning pada game Catur Cina (XiangQi). Tujuannya adalah sejauh mana waktu yang digunakan oleh kedua algoritma tersebut efektif dalam permainan Catur Cina. Metodologi yang digunakan dalam membangun aplikasi adalah Rapid Application Development, yaitu merupakan pengembangan dari metodologi Software Development Life Cycle. Kegiatan yang dilakukan antara lain yaitu melakukan perencanaan dan analisis terhadap pengembangan game dan melakukan pembuatan game dengan menggunakan game engine Unity dan bahasa pemrograman C#, Editor yang digunakan adalah Atom. Hasil pembuatan game dan koding algoritma akan di uji coba dengan iterasi kedalaman dan preset yang ditentukan sesuai dengan Minimax dan Alpha-Beta Pruning. Data yang didapat yaitu kecepatan dan banyak putaran antara kedua algoritma. Data tersebut akan dibandingkan sehingga performa kedua algoritma akan terlihat jelas.

Kata kunci—Chinese Chess, Board Game, Unity, MiniMax, AlphaBeta

Abstract

Algorithm is a complete and logical sequence, with a logical order in many ways done in a different order. In this case, the performance of the Minimax and Alpha Beta Pruning algorithms will be compared in the Chinese Chess game (XiangQi). The aim is to what extent the time spent by the two algorithms is effective in Chinese chess. The methodology used in building the application is Rapid Application Development, which is a development of the Software Development Life Cycle methodology. Activities carried out include planning and analyzing game development and making games using the Unity game engine and the C # programming language. The editor used is Atom. The results of game creation and algorithm coding will be tested with iteration depths and presets determined according to Minimax and Alpha-Beta Pruning. The data obtained are the speed and number of turns between the two algorithms. The data will be compared so that the performance of the two algorithms will be clearly seen.

Keywords— Chinese Chess, Board Game, Unity, MiniMax, AlphaBeta.

1. PENDAHULUAN

Game di dunia komputer sudah menjadi bagian dari perangkat lunak yang banyak digemari oleh pemakai komputer. Umumnya perkembangan sebuah komputer akan melalui perkembangan grafis yang terus mendorong kekuatan hardware grafik komputer maupun perkembangan *Artificial Intelligence* yang diterapkan pada *game-game* tersebut. Hal yang tidak dapat dilupakan pada *game-game* papan sederhana yang telah lama menjadi tempat kompetisi algoritma yang dituangkan dalam bentuk digital, Salah satunya adalah catur *Xiangqi*, *game* ini tidaklah berbeda dengan *game* papan lainnya walaupun *game* ini lebih langka daripada Catur Internasional.

Jaman modern ini *game* papan masih diminati dikalangan anak muda. *Game* Papan adalah *game* yang biasanya terdiri dari papan dan berbagai jenis potongan (dadu, bidak, counter, dll.), sistem aturan, dan dua atau lebih pemain. Dengan adanya bantuan teknologi, *game* papan tersebut beralih menjadi *game* digital.

Game Digital adalah program interaktif untuk satu atau lebih pemain, yang merupakan adaptasi sistem *game* 'tradisional', dengan aturan, representasi pemain, dan lingkungan yang dikelola melalui badan elektronik, Karena keterbatasan teknologi, *game* digital terdahulu sangatlah jarang yang memiliki fitur multiplayer, maka diharuskan memiliki *Artificial Intelligence* atau yang disebut secara singkat dengan istilah "AI".[1]

Artificial Intelligence adalah serangkaian algoritma yang luas yang juga mencakup teknik dari teori kontrol, robotika, grafik komputer, dan ilmu komputer secara umum, AI pada *game* digital sering tidak merupakan "AI" yang serupa pada penelitian ilmiah karena AI pada *game* digital jarang memfasilitasi pembelajaran komputer atau kriteria standar lainnya, hanya merupakan perhitungan otomatis atau serangkaian respons yang telah ditentukan dengan bantuan Algoritma-algoritma tertentu.[2]

Algoritma adalah prosedur atau formula untuk menyelesaikan masalah berdasarkan apa yang konduktif pada urutan tindakan tertentu. Program komputer dapat dilihat sebagai gabungan berbagai algoritma yang rumit. Dalam matematika dan ilmu komputer, algoritma biasanya berarti prosedur kecil yang memecahkan masalah berulang-ulang. Salah satu Algoritma yang sering dipakai pada *game* papan digital ialah Algoritma *Alpha-Beta Pruning*.

Algoritma *Alpha-Beta Pruning* adalah teknik optimisasi untuk algoritma minimax agar dapat mengurangi waktu perhitungan dengan lebih cepat. Algoritma tersebut memungkinkan sistem bekerja ke level yang lebih dalam di pohon *game* dan memotong cabang di pohon *game* yang tidak perlu dicari karena sudah ada langkah yang lebih baik.[2]

Algoritma Alpha-Beta Pruning dan MiniMax merupakan algoritma AI yang paling sering digunakan dalam *game-game* berbentuk papan seperti catur karena algoritma menggunakan analisis node pohon dalam memecahkan masalah.

Xiangqi, atau Catur Cina, adalah *game* yang sangat populer di negara-negara Asia. Dipercayai bahwa baik *Xiangqi* maupun Catur Barat/International telah dikembangkan dari *game Chaturanga* asli India.

Bidak catur Cina biasanya berbentuk cakram datar, memiliki nama-nama yang mirip dengan yang digunakan dalam catur barat dan tertulis di atasnya huruf Cina. Selain raja dan dua menteri, masing-masing pemain mulai dengan dua benteng, dua kuda, dua gajah, dua meriam, dan lima bidak.[3]

Dengan pengertian singkat tersebut jelas bahwa *Xiangqi* adalah *game* papan yang berbeda dengan catur pada umumnya, yaitu memiliki peraturan gerak yang berbeda, buah yang berbeda dan papan yang berbeda. Hal-hal tersebut membutuhkan algoritma *Alpha-Beta Pruning* yang berbeda dari catur seperti pada umumnya.

2. METODE PENELITIAN

Dalam melakukan penelitian ini, metodologi yang digunakan yaitu metodologi iterasi (Iteration). Dan Rapid Application Development (RAD).[4] Metode Iterasi memiliki fase pengembangan, yaitu sebagai berikut:

2.1 Tahapan Design

Dalam tahap ini dilakukan Studi Literatur, Penulisan batas-batas skenario dan desain awal sistem.

2.1.1 Studi Literatur

Studi literatur dalam sebuah penelitian untuk mendapatkan gambaran yang menyeluruh tentang apa yang sudah dikerjakan orang lain dan bagaimana orang mengerjakannya, kemudian seberapa berbeda penelitian yang akan dilakukan. Studi literatur penting untuk menghindari usaha yang sebenarnya sudah pernah dilakukan orang lain. Dalam penelitian ini, penulis mencari dan mempelajari literatur-literatur berupa jurnal yang berkaitan dengan masalah yang diteliti.

2.1.2 Penulisan Skenario

Penelitian mensimulasikan keadaan tertentu pada papan catur, yang selebihnya di panggil “Skenario” sebanyak 10 skenario masing-masing dengan 5 kedalaman node untuk mengetahui performa algoritma pada *game* catur cina.

2.1.3 Desain Sistem

Pada fase ini dibuat rancangan bagaimana program bekerja sesuai dengan alat ukur pengujian.

2.2 Tahapan Implementasi

Dalam tahap ini dilakukan Development & Demonstration terhadap algoritma serta program catur cina.

2.2.1 Development

Dalam tahap ini algoritma di implementasikan pada sistem agar program dapat berjalan sesuai dengan kebutuhan. Algoritma *MiniMax* dan *Alpha-Beta Pruning* diimplementasikan pada tahap ini.

2.2.2 Refine & Demonstrate

Pada fase ini dilakukan pengujian program dan algoritma. Program yang telah dibuat diuji dengan tindakan seperti meletakkan bidak pada skenario, melakukan gerakan dan menjalankan algoritma tertentu serta memastikan semua gerakan bidak/pion sesuai dengan peraturan gerak *game* pada Federasi Catur Cina Internasional

2.3 Tahapan Playtest

Pada fase ini penelitian melakukan analisis dan pengetesan hasil. Penulis membandingkan hasil performa dari kedua algoritma *MiniMax* dan *Alpha-Beta Pruning* berdasarkan kecepatan gerak, banyak node yang dilewati dan *Win-to-Lose Ratio* dengan mengukur skor menang dan kalah antara kedua sisi dengan algoritma yang berbeda

2.4 Tahapan Evaluate

Pada fase ini penelitian mengevaluasi hasil yang didapat dari fase *Playtest* dan membuat laporan.

2.5 Spesifikasi Perangkat Keras dan Lunak

Berikut ini adalah perangkat keras dan lunak yang digunakan selama ujicoba algoritma, diantaranya:

2.5.1 Perangkat Keras

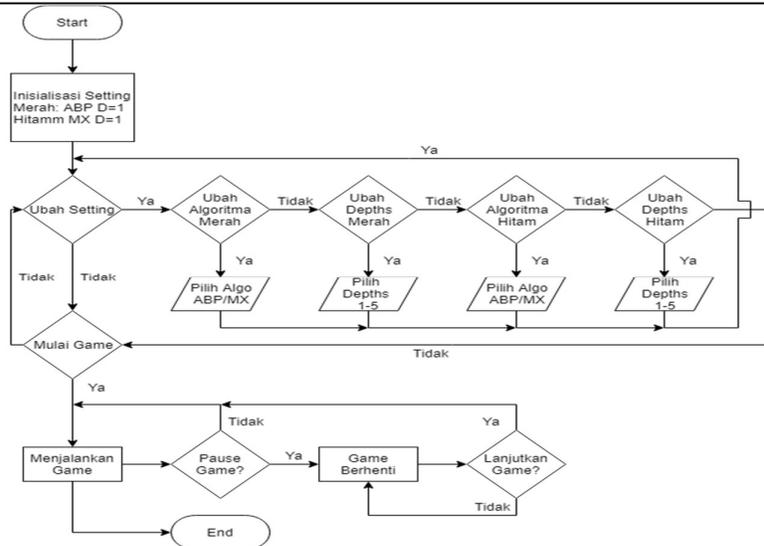
1. Laptop ASUS X540LJ dengan Processor Intel® Core™ i3-4005U CPU @ 1.70GHz x64 dan Random Access Memory 4.00 GB.
2. Desktop PC dengan Intel® Core™ i3-6100 CPU @ 3.70GHz x 64 dan Random Access Memory 8.00 GB

2.5.2 Perangkat Lunak

1. Sistem Operasi *Microsoft Windows 10*
2. *Game Engine Unity 2018.2.1f1 (64-bit)*.
3. Editor Program *Atom*

2.6 Flowchart Game

Untuk menggambarkan urutan-urutan proses pada system ini, maka digunakan sebuah flowchart sebagai penjelas. berikut ini adalah flowchart yang menggambarkan mulai *game*

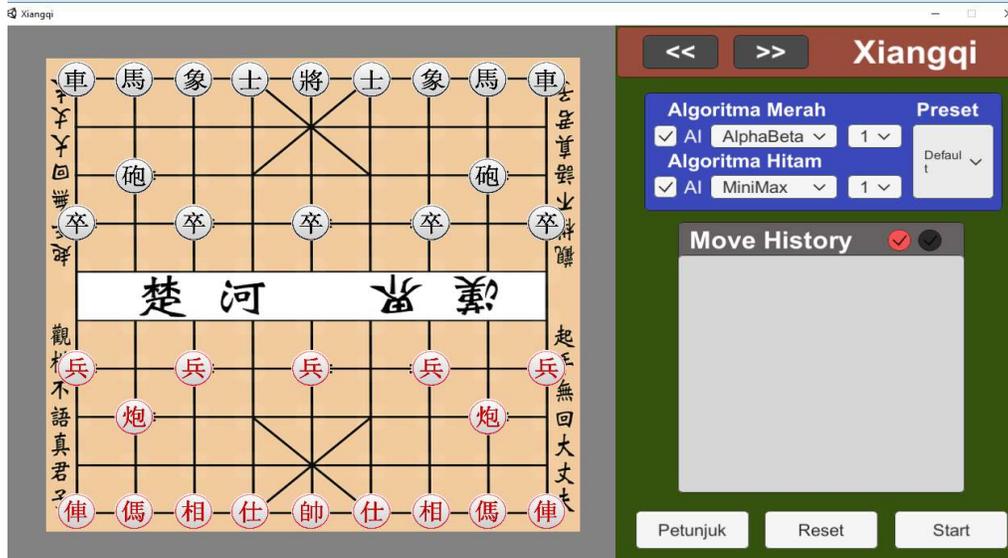


Gambar 1 Flowchart Game

3. HASIL DAN PEMBAHASAN

3.1 Tampilan Game

Berikut ini adalah tampilan game secara keseluruhan.



Gambar 2 Tampilan Game Catur Cina

3.2 Tabel Hasil

Dengan terkumpulnya data performa algoritma Minimax dan Alphabeta pada masing-masing *Preset* papan Xiangqi, maka tabel dapat disusun sesuai algoritma dan sisi papan, sehingga didapat hasil rata-rata pada kecepatan dan jumlah putaran.

Tabel 3.1 Rata-rata Minimax Sisi Hitam

Black Minimax	1	2	3	4	5
Default	0.042	0.038	17.210	92.420	
Pawn Off The River A	0.072	0.188	2.291	46.120	
Pawn Off The River B	0.005	0.027	0.148	5.206	
Cannons Ommited	0.006	0.099	1.037	10.113	
Flanking Cannons	0.031	0.217	3.895	135.090	
Cannons and Forts	0.017	0.310	7.385	55.260	
Knights and Elephants	0.016	0.038	0.233	3.930	
Castle Siege A	0.003	0.158	0.789	33.683	
Castle Siege B	0.006	0.157	1.470	40.704	
Mirrored Advances	0.013	0.330	3.301	59.070	
Average	0.021	0.156	3.776	48.160	

Tabel 3.2 Rata-rata Minimax Sisi Merah

Red Minimax	1	2	3	4	5
Default	0.042	1.000	35.420	159.590	
Pawn Off The River A	0.015	0.361	6.182	95.670	
Pawn Off The River B	0.006	0.103	1.046	17.488	
Cannons Ommited	0.006	0.103	1.369	8.661	
Flanking Cannons	0.015	0.395	10.539	248.740	
Cannons and Forts	0.025	0.422	9.403	54.010	
Knights and Elephants	0.016	0.039	0.235	4.352	
Castle Siege A	0.008	0.219	3.626	63.369	
Castle Siege B	0.020	0.257	3.203	56.721	
Mirrored Advances	0.014	0.536	4.394	69.270	
Average	0.017	0.344	7.542	77.787	

Tabel 3.3 Rata-rata AlphaBeta Sisi Hitam

Black Alphabeta	1	2	3	4	5
Default	0.047	0.547	17.269	127.120	
Pawn Off The River A	0.005	0.182	2.297	46.040	
Pawn Off The River B	0.005	0.030	0.144	5.019	
Cannons Ommited	0.003	0.102	1.503	9.140	
Flanking Cannons	0.017	1.088	3.328	75.360	
Cannons and Forts	0.011	0.166	7.762	64.840	

Knights and Elephants	0.002	0.038	0.226	87.910	
Castle Siege A	0.004	0.157	1.345	26.473	
Castle Siege B	0.012	0.150	1.166	39.860	
Mirrored Advances	0.032	0.152	3.394	79.870	
Average	0.014	0.261	3.843	56.163	

Tabel 3.4 Rata-rata AlphaBeta Sisi Merah

Red Alphabeta	1	2	3	4	5
Default	0.045	0.732	17.747	190.600	
Pawn Off The River A	0.012	0.357	6.169	95.850	
Pawn Off The River B	0.011	0.069	0.756	21.427	
Cannons Ommited	0.004	0.104	1.367	8.669	
Flanking Cannons	0.015	0.404	10.699	219.190	
Cannons and Forts	0.011	0.338	16.102	116.720	
Knights and Elephants	0.004	0.050	0.327	5.695	
Castle Siege A	0.007	0.212	4.054	54.880	
Castle Siege B	0.023	0.344	3.060	55.070	
Mirrored Advances	0.056	0.377	4.660	57.140	
Average	0.019	0.299	6.494	82.524	

Tabel 3.5 Rata-rata Putaran Minimax

Turns Minimax	1	2	3	4
Default	19	10	9	7
Pawn Off The River A	3	3	3	3
Pawn Off The River B	3	3	3	10
Cannons Ommited	7	7	7	7
Flanking Cannons	9	4	4	5
Cannons and Forts	9	5	4	2
Knights and Elephants	5	5	4	5
Castle Siege A	3	5	5	8
Castle Siege B	9	8	7	19
Mirrored Advances	5	4	3	3
Average	7.2	5.4	4.9	6.9

Tabel 3.6 Rata-rata Putaran AlphaBeta

Turns Alphabeta	1	2	3	4
Default	18	10	9	10

Pawn Off The River A	3	3	3	3
Pawn Off The River B	12	5	5	11
Cannons Omitted	7	7	7	7
Flanking Cannons	6	6	3	3
Cannons and Forts	4	3	5	3
Knights and Elephants	5	5	5	10
Castle Siege A	5	7	8	6
Castle Siege B	9	9	7	19
Mirrored Advances	14	3	3	4
Average	8.3	5.8	5.5	7.6

Tabel 3.7 Rata-rata Kecepatan Algoritma

Average	1	2	3	4	5
Red MiniMax	0.017	0.344	7.542	77.787	
Red Alphabet	0.019	0.299	6.494	82.524	
Black MiniMax	0.021	0.156	3.776	48.160	
Black Alphabet	0.014	0.261	3.843	56.163	

4. KESIMPULAN

Perbandingan Performa *Alpha-Beta Pruning* dan *MiniMax* pada Catur Cina (*XiangQi*) ini dimulai dengan perencanaan, desain dan implementasi, uji coba dan kemudian evaluasi. Dari proses-proses tersebut, penelitian ini memiliki kesimpulan sebagai berikut :

1. Dapat dibuktikan bahwa algoritma *Alpha-Beta Pruning* tidak lebih cepat secara waktu atau efisien dalam banyak putaran dari algoritma *MiniMax* pada *game* catur cina (*Xiangqi*).
2. Perbedaan *platform* ujicoba yang dispesifikasikan di bab 3 dapat mempengaruhi kecepatan gerak algoritma, tetapi hasil yang didapat pada *platform* Laptop konsisten jika dibandingkan dengan hasil uji coba *platform* Desktop dengan rasio 2:1.
3. Suksesnya implementasi *game* Catur Cina (*XiangQi*) berbasis desktop dengan algoritma *Alpha-Beta Pruning* dan *MiniMax* serta 10 preset dan mode Player vs AI atau AI vs AI.

5. SARAN

Dalam pengembangan dan perbandingan *Alpha-Beta Pruning* dan *MiniMax* pada Catur Cina (*XiangQi*) dapat disadari masih banyak kekurangan yang terdapat pada game serta penelitian yang telah dilakukan, Oleh sebab diharapkan penelitian ini dapat dikembangkan secara lebih lanjut. Saran penelitian ini yaitu :

1. Penelitian yang lebih lanjut mengenai *E-value*, yaitu bobot yang dimiliki setiap bidak sehingga algoritma dapat memproses gerakan bidak yang lebih pintar.
2. Penerapan atau pengembangan algoritma yang berbeda pada game *Xiangqi* agar mendapat *AI* yang lebih cepat dan efektif.
3. Penerapan pergerakan bidak serta logika koding yang lebih efisien, termasuk berbagai peraturan *Xiangqi* khusus dan pengecekan *check* dan *checkmate* yang sesuai.

DAFTAR PUSTAKA

- [1] G.N. Yannakakis. 2012, *Game AI Revisited*. Center for Computer Games Research IT University of Copenhagen, Denmark.
- [2] R. Stuart, N. Peter. 2018, *Artificial Intelligence: A Modern Approach*, Pertinence Hall. New Jersey.
- [3] Yen, Shi-Jim Chen, JC Yang, TN Hsu, Shun-Chin, 2004, Computer Chinese Chess. *ICGA journal*. 27. 3-18. 10.3233/ICG-2004-27102. Taiwan.
- [4] Hidayatullah, Priyanto, Jauhari Khairul Kawistara, 2014, Pemrograman WEB. Bandung : Informatika Bandung. (jQuery)