INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

Design of Checkers Game Using Alpha-Beta Pruning Algorithm

Perancangan Game Checkers Menggunakan Algoritma Alpha Beta Pruning

Received:
19 March 2021
Revised:
31 July 2021
Published:
8 August 2021

[1*]Achmad Naufal Wijaya Jofanda, [2]Mohamad Yasin
[1,2]Matematika, Universitas Negeri Malang
[1,2]Malang, Indonesia
E-mail: [1]achmadnaufal.1703126@students.um.ac.id,
[2]mohamad.yasin.fmipa@um.ac.id
*Corresponding Author

Abstract— Checkers is a board game that is played by two people which has a purpose to defeat the opponent by eating all the opponent's pieces or making the opponent unable to make a move. The sophistication of technology at this modern time makes the checkers game can be used on a computer even with a smartphone. The application of artificial intelligence in checkers games makes the game playable anywhere and anytime. Alpha Beta Pruning is an optimization technique from the Minimax Algorithm that can reduce the number of branch/node extensions to get better and faster step search results. In this study, a checkers game based on artificial intelligence will be developed using the alpha-beta pruning method. This research is expected to explain in detail how artificial intelligence works in a game. Alpha-beta pruning was chosen because it can search for the best steps quickly and precisely. This study tested 10 respondents to play this game. The results show that the player's win rate was 60% at the easy level, 40% at the medium level, and 20% at the hard level. Besides that, the level of interest in this game was 80% being entertained and 20% feeling ordinary.

Keyword— checkers, artificial intelligence, alpha-beta pruning

Abstrak— *Checkers merupakan pemainan papan (board game) yang dimainkan oleh 2 orang yang bertujuan mengalahkan lawan dengan cara memakan semua bidak lawan atau membuat lawan tidak dapat melakukan gerakan. Kecanggihan teknologi pada zaman ini membuat permainan checkers dapat dilakukan dengan menggunakan komputer bahkan dengan smartphone sekalipun. Penerapan artificial intelligence pada checkers game membuat game tersebut dapat dimainkan dimana saja dan kapan saja. Alpha Beta Pruning adalah suatu teknik optimasi dari Algoritma Minimax yang dapat mengurangi jumlah perluasan cabang/node untuk mendapatkan hasil pencarian langkah yang lebih baik dan cepat.Dalam penelitian ini akan dikembangkan sebuah checkers game yang berbasis artificial intelligence dengan menggunakan metode alpha beta pruning. Dengan adanya penelitian ini diharapkan mampu menjelaskan secara detail bagaimana artificial inteligence bekerja dalam sebuah game.Alpha beta pruning dipilih karena mampu melakukan pencarian langkah terbaik dengan cepat dan tepat. Penelitian ini melakukan pengujian terhadap 10 responden untuk memainkan game ini. Didapatkan hasil bahwa tingkat kemenangan pemain adalah sebesar 60% pada tingkat easy, 40% pada tingkat medium, dan 20% pada tingkat hard. Selain itu tingkat kemenarikan pada game ini adalah 80% terhibur dan 20% merasa biasa.*

Kata Kunci— *checkers, artificial intelligence, alpha-beta pruning*

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

I. INTRODUCTION

A game is an activity that has several complex rules in it that aim to give happiness [1]. According to KBBI, a game is something that is used to play or just to have fun. In a game, several rules must be followed by players to determine the outcome. The media for the game is also very diverse, some games use boards and pawns such as chess, monopoly, checkers, snakes, and ladders, and so on. In addition to board games, there are also that needed balls such as soccer, volleyball, basketball, and so on. The term is often used for the game based on technology. There are several types of games based on the platform of the game, including Arcade Games, PC Games, Console Games, Mobile Games, and so on. Based on the genre, games also have various types such as puzzles, sports games, Role Playing Games (RPG), education, simulation, strategy, adventure, and so on. A lot of types of games make the game very popular among a lot of people.

In-game development, developers always consider the algorithm to be used, therefore the results of the game become more fun and interesting. Developers have done various ways until one day in 1956, John McCarthy and three of his friends worked on a project for two months at Dartmouth. The project produced a concept that made a program capable of non-numerical thinking, known as Principia Mathematica. This concept became the forerunner to the formation of the concept of artificial intelligence or what is commonly called artificial intelligence.

Artificial Intelligence (AI) is a program design that allows computers to make decisions by imitating human thinking and reasoning[2][3]. In its application, Artificial Intelligence can be implemented in many fields, including in games. The application of Artificial Intelligence to this game aims to provide thinking skills on computers that can equal human thinking abilities[4]. With the AI that works in a game, it makes the game more interesting to play because players can play using a computer or machine. The development of its very advanced technology makes the sophistication of AI in computers as opposed to playing in games also getting smarter in making decisions [5]. AI has a role in the game as a means of determining game strategy. It is because there is an algorithm that works on AI as a decision making, therefore it can choose the best decision in preventing players from winning [6]. Several kinds of fields that use Artificial Intelligence include expert systems, computer games (games), fuzzy logic, artificial neural networks, and robotics [7].

Checkers is a board game played by two people with several rules contained in it. The objective of the game is to beat the opponent or win the game by eating all the opponent's pieces or making the opponent unable to make a move[8]. Checkers was first played around 3000

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

before the century. At that time the medium of the game was slightly different from the checkers that are known today. Checkers are played on a board with a size of 8 × 8 squares with 12 pieces for each player.

The sophistication of technology at this time makes the checkers game can be done using a computer even with a smartphone. But what about the opponent in the game? With AI, the checkers game can be done by making the computer the opponent of the game. It is often called a bot. The use of these bots makes the checkers game even more interesting. This game can be played anywhere and anytime because the concept of this bot game is to use a human point of view and human game behavior, so there is no need to use humans to play opponents [9].

The use of decision-making methods in artificial intelligence allows games to be played alone. The automatic steps that the computer takes when playing is a form of implementing artificial intelligence decision-making that is on the computer [10]. Many decision-making methods have been applied to various kinds of problems, one of them is Alpha-Beta Pruning. Alpha Beta Pruning is an optimization technique from the Minimax Algorithm that can reduce the number of branch/node extensions to get better and faster step search results[11]. This algorithm is a search algorithm that is used to reduce the number of nodes through the evaluation process [12]. The application of Alpha-Beta Pruning in this checkers game is as a computer decision making with the main input being human perception. The meaning of human perception here is the step that the player chooses in carrying out the pawn so that the computer can make the best decisions in carrying out its own steps based on the player's steps.

Several previous studies that used the Alpha Beta Pruning method in games were researched on the Analysis of the Effectiveness of Alpha-Beta Pruning, Minimax, and Negamax on a board game [13]. In his article, it is said that the Alpha Beta Pruning algorithm is quite effective in using the 10 game titles that he chooses or rather has 50% effectiveness on the 10 titles [13]. Then from the research also obtained results from 40 game titles, Alpha Beta Pruning has a match of 45% and 15% for negamax . In addition, there is other research on the comparison of the Minimax and Alpha-Beta Pruning algorithms in the making of early games [14]. This article stated that Alpha Beta Pruning has an advantage in the speed of execution time. All in all, it can be concluded that Alpha Beta Pruning still has advantages in terms of time efficiency and effectiveness. Therefore, the authors want to try implementing the Alpha Beta Pruning method to see how effective and efficient this method is in determining the step decisions of the computer. In addition, the authors also want to implement the Alpha Beta Pruning algorithm on the effectiveness of the checkers game decision-making step.

Based on the explanation above, the aim of this study is to combine the checkers with the Alpha Beta Pruning algorithm. Alpha Beta Pruning is a search algorithm that is formed from the

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

Minimax Algorithm which has been optimized so that the time needed to search will be reduced. This algorithm is a search algorithm that is used to reduce the number of nodes through the evaluation process [12]. Alpha Beta Pruning is often used in making board games. Some examples include games tic-tac-toe, chess, checkers, and many more. Because this algorithm is an optimization of the Minimax Algorithm, the search process is the same as Minimax. The difference between this algorithm and Minimax lies in two additional parameters, called alpha and beta. Alpha is the best step that can be ensured by the maximizer when the current or previous state is in the maximizer's turn. Meanwhile, beta is the best value that can be ascertained by the minimizer when the current state or before the minimizer is rotated [15].

The algorithm is applied to the checkers game to make enemy steps more varied and can make moves based on pawn steps carried by the player. The benefits that will be obtained in making this application are providing an understanding of the implementation of artificial intelligence in checkers games using AI, as well as the Alpha Beta Pruning algorithm in choosing the right steps when playing.

II. RESEARCH METHOD

This research is considered as development research by applying the waterfall model of research suggested by Winston W. Royce. The motive for applying this method is that it describes steps that are structured and sequential [16]. The following steps are:

1. Requirements

The requirements stage or determining system requirements is an analysis of system requirements which is carried out to understand what is expected and the limitations of the application to be made. At this stage, information searches are carried out regarding how to make applications, the problems that will be resolved, and the methods used, and so on. In this process, it takes several things that have to exist when creating a program. Some of the above things reference in the manufacture of checkers game with python references about how the Alpha Beta Pruning algorithm, several references Pygame, as well as references to checkers attractive design.

2. Design

The design stage or map-out design is carried out by the results of the analysis that has been carried out previously at the requirements stage. This checkers game is made by the Python programming language with the help of PyGame that can be used to create Python-based games. This application has 3 kinds of pages. The first page is the main menu which contains a button to start the game, a how-to-play button, and an exit button. The second page is the game difficulty

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

level selection page. The third page is the page for the rules and how to play checkers. The last page is the game page that contains the game.

3. Implementation

Implementation of application is made by designs that have been made previously. This application is created using Python language or precisely using PyGame which is one of the modules from Python. The software used in creating this application is with the help of Visual Studio Code software. This stage is considered the main stage in creating the application.

4. Verification

Verification of application testing is carried out to know application performance and compliance with system requirements analysis. Moreover, this stage is also used to determine whether the implementation stage is accurate or not. If an error occurs, the process turns back to the implementation stage and replicates the verification stage. The triumph of the verification process is marked by the success or failure of the program created, the selection of the right computer steps, and the level of difficulty of the player when playing the game. The method used to verify is the black-box testing method. Black-box testing is a test based on application details such as the appearance of the application, the functions in the application, and the suitability of the function flow. This test is done by trying each of the functions contained in the game by seeing whether the output generated when the function is executed is as expected. In addition, testing is also carried out by the usability testing method or testing of several respondents to play this game. In this test, 10 respondents were taken with various backgrounds such as students, workers, or non-workers. Furthermore, the age ranges of these 10 respondents also varied from under 20 years of age to over 20 years. Respondent selection is also sought based on their knowledge of checkers so that this study will produce various results.

5. Maintenance

After verification or testing, the next step is application maintenance. Apart from application maintenance, application development is also carried out over time as needed. The form of application development is the addition of new features so that the application becomes better. This stage is also the last and it is always carried out over time and needs.

Data Collection and Problem Analysis

Humans are grown up by making mistakes, evaluating mistakes, and fixing those mistakes. Grievously, it is not an effective way in a short time, it takes plenty of evaluation and improvement to create an effective and satisfying result. It is different with computers. The computer is not a very long and time-consuming process. The computer does not think like humans, it performs calculations with several formulation techniques that can produce the best

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

decisions to be taken. Similarly, when playing the checkers game, intelligence in the computer can easily overpower great checkers players and experienced only by a few minutes due to the way of thinking computers that are effective and efficient in determining the pace. Currently, several search algorithms are very commonly used in determining the best move. Some of them are minimax, alpha-beta pruning, negamax, negascout, and so on.

This study used the alpha-beta pruning algorithm method to determine the best step. The reason for using this algorithm is because this algorithm can analyze all possible game steps from the computer itself or the player. In addition to this, the alpha-beta pruning algorithm can work quickly and precisely because it uses the pruning method or tree cutting so that the time required is shorter. With the use of this algorithm, the depth of the search can be determined by yourself so that the computer's ability will be better as well. The difficulty level of the player against the computer also becomes more difficult because the computer can think for the next few steps depending on the depth used.

III. RESULT AND DISCUSSION

A. System Design

This design focuses on the user interface or game design so that the game will feel more attractive when played.

1. User Interface

A good user interface and easy-to-understand are needed when someone is going to run an application. With an easy-to-use user interface, users will easily understand how to operate the application properly

2. Main Menu Design

This main menu design is a page that will be displayed when the user starts a new game. This form has 2 pages, namely the initial main menu and the difficulty level main menu. On the main menu page, 3 kinds of buttons are given, namely play, how to play, and quit. When the play button is clicked, a new page containing 3 kinds of buttons, namely easy, medium, and hard where each button represents the difficulty level of each will be loaded. When the how to play button is clicked, a new page containing the checkers game rules will be loaded.

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

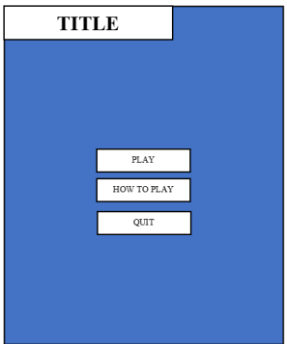

Figure 1. MAIN MENU PAGE DESIGN

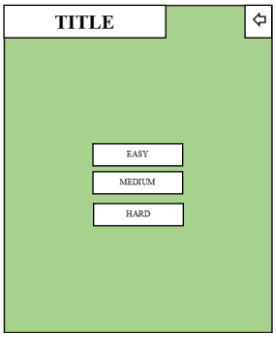

Figure 2. DIFFICULTY OPTION PAGE DESIGN

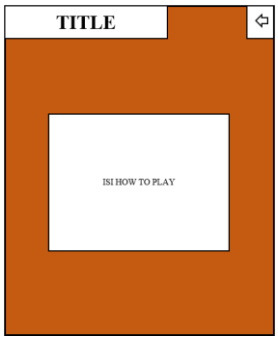

Figure 3. HOW TO PLAY PAGE DESIGN

3. Game Interface Design

The interface game design is a page that is loaded when the user has clicked on the difficulty level option. This page contains a black and yellow 8 × 8 checkers board with 12 checkers on each side, white for the computer and red for the player. There are 2 additional buttons below the checker's board which function as the undo button and the home button. The game interface design can be seen in Figure 4.

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

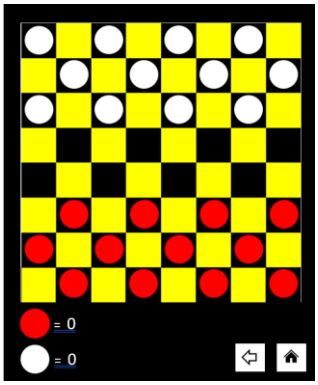

Figure 4. INTERFACE GAME DESIGN

B. Analysis and System Design

In the system, the computer is programmed to fight the user, the user can also choose the level of difficulty and choose to exit the game. This is because the user is in charge as a player and as a person who runs the application. When the user starts running the application, the flow carried out by the user is as shown in Figure 5 below.

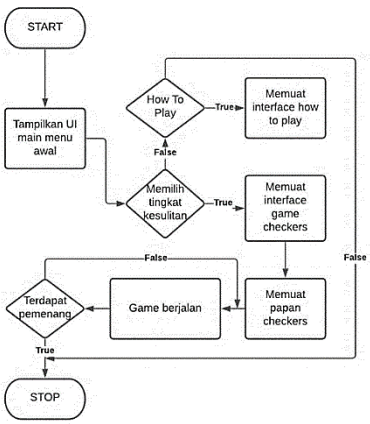

Figure 5. GAMEPLAY FLOWCHART

C. Interface Design Implementation

The following are the components contained in the checkers game.

1. Main Menu Page Display

This page displays 3 kinds of buttons play, how to play, and quit which have different functions. The play button is a button that functions to start the game, the how to play button is a button to go to the how-to-play page and the quit button is a button with the function of exiting the game. Figure 6 is an illustration of the main menu page.

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

Figure 6. MAIN MENU PAGE

2. How to Play Page Display

This page displays a pair of illustration images and descriptions of these explanations. This display consists of 7 pages, each of which contains different illustrative images with different descriptions according to the explanation. Figure 7 is a display example of how to play

Figure 7. HOW TO PLAY DISPLAY

3. Difficulty Option Page Display

This page, shown a display of difficulty options that can be selected by the player. Each of these options has its respective functions. There are 3 kinds of options, that are "Easy", "Medium", and "Hard". Each difficulty option represents its own difficulty level. The difference from each of these options is in the depth used by the working Artificial Intelligence. The depth of "Easy" is 1, "Medium" is 2, and "Hard" is 3. Figure 8 below is a display of the difficulty option page.

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

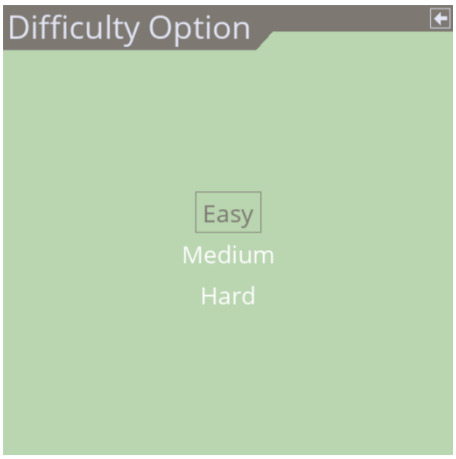

Figure 8. DIFFICULTY OPTION PAGE

4. Game Display

This page showed a display of the checker's board and its pieces. This view is the main view of the checkers game. Figure 9 below is a display of the game.

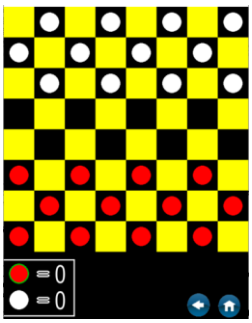

Figure 9. GAME DISPLAY

In the study process, 2 kinds of tests were carried out. The first one is the black-box testing of the checkers game. The test conducted in this section is to see the ability of the alpha-beta pruning algorithm in executing steps from the computer. This trial was conducted on an Asus X550IU PC that has the Windows 10 operating system and 16 GB RAM. The second test is usability testing or gameplay testing by asking 10 respondents with different backgrounds to play the game.

1. Black-Box Testing

Black box testing is done to see if any bugs occur when checkers games are played [17]. The results show that every feature in the application is running smoothly and without problems. The results of black-box testing can be seen in Table 1.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

Table 1. BLACK-BOX TESTING RESULTS

Test No	*Test Function*	*Output*	*Result*
Test-1	*Select Piece*	*Valid-moves* appear	*Valid*
Test-2	*Select Blank Square*	No *valid-moves* appear	*Valid*
Test-3	*Move Piece to Blank Square*	*Piece's* location changed	*Valid*
Test-4	*Move Piece to Square with Piece*	The selected *piece* doesn't change/move	*Valid*
Test-5	*Jump over some piece*	*A piece* that was jumped on disappears	*Valid*
Test-6	*Repeat Game (yes)*	*The game* can be played again	*Valid*
Test-7	*Repeat Game (close)*	The *game* stays on the last page	*Valid*
Test-8	*Exit (yes)*	The *game* ends and the application stops	*Valid*
Test-9	*Exit (no)*	The *game* keeps running and the display stays	
Test-10	*Main Menu (difficulty option)*	The display changes when the *easy, medium,* and *hard* options are selected	*Valid*
Test-11	*Computer's Move*	*The piece* moves well with smart playing strategies	*Valid*
Test-12	*Undo Button*	The *board* view changes to the previous step	*Valid*
Test-13	*Home Button*	The page changes to the difficulty level option	*Valid*

From the 13 function tests, it was found that each function went well and as expected.

2. Gameplay Testing / Usability Testing

Usability testing was carried out by asking 10 respondents with different backgrounds (as shown in Table 2) to play the checkers game in 3 difficulty levels. The results obtained are as in Table 3.

Table 2. DETAILS OF THE RESPONDENT'S BACKGROUND

No	***Details of The Respondent's Background***	
	Type of Data	***Information***
1	Age	1 Person aged 16 years 4 People aged 21 years 3 People aged 22 years 1 Person aged 23 years 1 Person aged 26 years
2	Job Status	7 People are students 2 People are worker 1 Person, not a worker
3	Knowledge of *checkers*	10 People know *Checkers*
4	Experience playing *checkers*	2 People never play *Checkers* 8 People played *Checkers*
5	Proficiency level from a scale of 1 to 10	3 People in scale 1 1 Person on a scale of 3 2 People on a scale of 5 1 Person on a scale of 6 2 People on a scale of 7 1 Person on a scale of 8

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

Table 3. PLAYER'S WIN RATE PERCENTAGE

No	*Player's Win Rate Percentage*		
	Difficulty level	***W-D-L (Player%)***	***Keterangan***
1	"*Easy*"	6-0-4 (60%)	10 Game by 10 people
2	"*Medium*"	4-0-6 (40%)	10 Game by 10 people
3	"*Hard*"	2-0-8 (20%)	10 Game by 10 people

From Table 3, it can be concluded that the player's win rate at each difficulty level decreases because the depth used is higher. The result is that computers are getting smarter because they can track a greater number of the best steps.

Table 4. USER INTERFACE DESIGN, EASE OF APPLICATION, DAN GAMEPLAY RESULT TESTS

No	**Questions**	**Satisfaction rate**			
	User Interface	**Interesting**	**Good**	**Ordinary**	**Not Good**
1	How does the overall appearance of the checkers game?	20 %	70 %	10 %	0 %
2	How does the display of mainmenu?	20 %	50 %	30 %	0 %
3	How does the display of board game?	20 %	40 %	30 %	10%
4	How does the display of the pieces?	0 %	50%	50 %	0 %
5	How does the display of "How to Play"menu?	50 %	50 %	0 %	0 %
Average		22 %	52 %	24 %	2 %
	Ease of Application	**Very easy**	**Easy**	**Difficult**	**Very difficult**
6	How easy is it to play the whole game of checkers?	20 %	60 %	20 %	0 %
7	How easy is it to access the checker's game menu?	30 %	70 %	0 %	0 %
8	How easy is it to move the game pieces?	20 %	80 %	0 %	0 %
Average		23,33 %	70 %	6,667 %	0%
	Gameplay (step variation)	**Unpredictable**	**Good**	**Predictable**	**Very predictable**
9	How does the computer step at the "Easy" level?	10 %	70 %	10 %	10 %
10	How does the computer step at the "Medium" level?	40 %	50 %	10 %	0 %
11	How does the computer step at the "Hard" level?	60 %	30 %	10 %	0 %
Average		36,667 %	50 %	10 %	3,33%

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

Table 4. USER INTERFACE DESIGN, EASE OF APPLICATION, DAN GAMEPLAY RESULT TESTS [CONTINUE]

No	Questions	Satisfaction rate			
	Gameplay (step speed)	**Very fast**	**Fast**	**Usual**	**Slow**
12	How fast does the computer move the pieces at the "Easy" level?	40 %	30 %	30 %	0 %
13	How fast does the computer move the pieces at the "Medium" level?	10 %	70 %	20 %	0 %
14	How fast does the computer move the pieces at the "Hard" level?	20 %	50 %	10 %	20 %
Average		23,33 %	50 %	20 %	6,667 %
	Attractiveness	**Very entertained**	**Entertained**	**Ordinary**	**Boring**
15	Are you entertained by checkers game?	0 %	80 %	20 %	0 %

From Table 4 above, the results show that game checkers are quite liked by 10 respondents because according to the data, 80% are entertained and 20% feel ordinary with game checkers. In addition, data is obtained on each question category, which are user interface, ease of application, and gameplay. The data explain that this game is quite good to play because as much as 70% on average of the total respondents said that this game is easy to play. Apart from that, in terms of computer games, 50% on average of the total respondents said that the movement of computers at each level was good and 36.66% said it was difficult to predict. In terms of speed, it is also found that on average 50% of the total respondents said that the computer made steps quickly and 23.33% said it was very fast. For more details, see Figure 10, Figure 11, Figure 12, and Figure 13 for the average graph of each category.

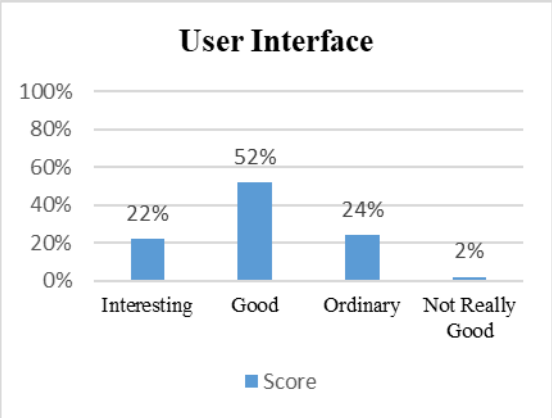

Figure 10. USER INTERFACE GRAPH RESPONSE

INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

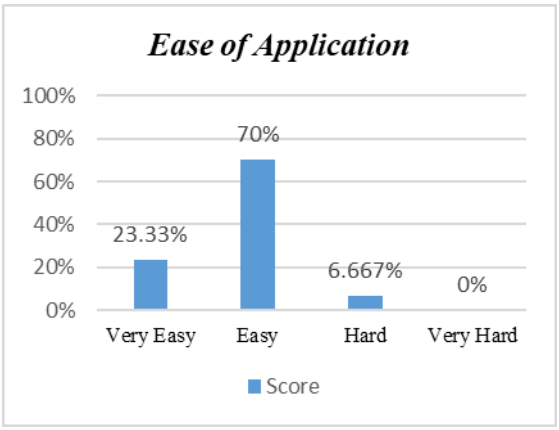

Figure 11. EASE OF APPLICATION GRAPH RESPONSE

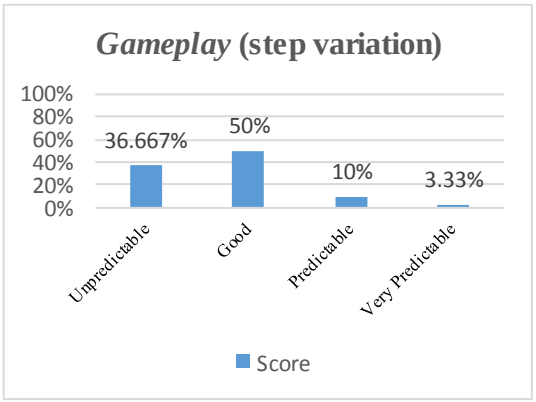

Figure 12. GAMEPLAY (STEP VARIATION) GRAPH RESPONSE

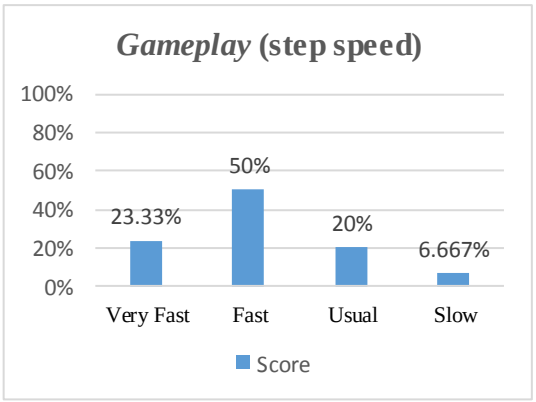

Figure 13. GAMEPLAY (STEP SPEED) GRAPH RESPONSE

From Figure 10 to Figure 13, it can be concluded that the game checkers application is quite interesting to play. The average answer from 10 respondents to the 5 questions given related to the user interface said that 52% of them chose good and 22% said it was very good. In addition, in terms of ease of running the application, the average answer to the 3 questions given said that

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

70% of them chose easy and 23.33% chose very easy. Apart from that, in terms of gameplay, it can be seen that computer games in terms of variations in the selection of steps and the speed of taking steps are already fairly good. From this, it can be concluded that the alpha-beta pruning algorithm is quite good to use in the step search implementation. Table 3 above, it can be seen that the use of the alpha-beta pruning algorithm is quite effective because it can outperform players by winning the game several times at each different level. A hard level can be seen that it has the least percentage of player wins which means the algorithm works well with the highest depth level

Many previous studies related to making checkers using several algorithms in artificial intelligence. For example, a study by [18] examines the implementation of the negascout algorithm in the checkers game. In addition, there is another research by [8] about the minimax algorithm on checkers and [19] about the implementation of the greedy algorithm in determining the piece steps in the checkers game. In the research that the author conducted by asking 10 people with different backgrounds, the results were as written in table 3. From these results, it can be concluded that the alpha-beta pruning algorithm is quite effective in determining the steps of the piece so that they can play against other players. In addition, with a large enough percentage of wins, it can be said to be better than the minimax algorithm because in a study conducted by [20] it was stated that the computer's winning rate using depth 3 (equivalent to the author's “Hard” level) gets the percentage of wins. only 50% while the author's alpha-beta pruning has an 80% wins rate for computers. In addition, if we can conclude from table 3, increasing the depth can affect the percentage of wins from computers because the level of computer intelligence is also higher. So, it can be concluded again that the alpha-beta pruning algorithm is Suitable for use as artificial intelligence that works in-game checkers.

IV. CONCLUSION

This research was conducted on the checkers game application based on artificial intelligence using the alpha-beta pruning method. The conclusion is that the game checkers application based on artificial intelligence (AI) using the alpha-beta pruning method works based on the user’s input who takes the first steps. The AI in this application works by simulating every possible step first. Each level in the checkers game has a depth level that is used to determine the best move. For easy level using depth 1, medium using depth 2, and hard using depth 3. From the research that was conducted on 10 respondents with different backgrounds, the results showed that 80% were entertained and 20% felt ordinary with checkers game. In this study, it was also found that the difficulty level in this game was quite difficult. The results obtained for easy difficulty only

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

get 60% of the player's win percentage, medium 40% of the player's win, and hard only 20% of the player's win. This game is made still in offline form so it cannot be accessed online yet. For further development, a checkers application can be made in an online form that contains the PVP (player versus player) feature, adding animation when the checkers pawn jumps against the opponent's checkers pawn.

REFERENCES

[1] F. Marzian dan M. Qamal, "Game Rpg 'The Royal Sword' Berbasis Desktop Dengan Menggunakan Metode Finite State Machine (Fsm)," J. Ilm. Sist. Inf. SISFO, vol. 1, no. 2, hlm. 36, 2017.

[2] I. P. W. Merta, I. M. G. Sunarya, dan I. K. R. Arthana, "Handgesture To Text Dengan Metode Artificial Intelligence KNN (K-Nearest Neighbour)," Kumpul. Artik. Mhs. Pendidik. Tek. Inform., vol. 4, no. 1, hlm. 10, 2015.

[3] D. A. Hashimoto, T. M. Ward, dan O. R. Meireles, "The Role of Artificial Intelligence in Surgery," Adv. Surg., vol. 54, hlm. 89–101, Sep 2020, doi: 10.1016/j.yasu.2020.05.010.

[4] C. Waruwu dan D. Purba, "Implementasi Algoritma Minimax Dalam Game Othello," J. Tek. Inform. Unika St Thomas JTIUST, vol. 2, no. 1, hlm. 11, 2017.

[5] M. F. Rahadian, A. Suyatno, dan S. Maharani, "Penerapan Metode Finite State Machine Pada Game 'The Relationship,'" Jurnal Informatika Mulawarman, vol. 11, hlm. 9, 2016.

[6] Y. Duan, J. S. Edwards, dan Y. K. Dwivedi, "Artificial Intelligence For Decision Making In The Era Of Big Data – Evolution, Challenges And Research Agenda," Int. J. Inf. Manag., vol. 48, hlm. 63–71, Okt 2019, doi: 10.1016/j.ijinfomgt.2019.01.021.

[7] M. Kurniawan, A. Pamungkas, dan S. Hadi, "Algoritma Minimax Sebagai Pengambil Keputusan Dalam Game Tic-Tac-Toe," Open Journal System SEMNASTEKNOMEDIA ONLINE, vol. 4, hlm. 6, 2016.

[8] D. Syapnika dan E. R. Siagian, "Penerapan Algoritma Minimax Pada Permainan Checkers," J. Ris. Komput. JURIKOM, vol. 2, no. 6, hlm. 28–32, 2015.

[9] G. Acampora, V. Loia, dan A. Vitiello, "Improving game bot behaviours through timed emotional intelligence," Knowl.-Based Syst., vol. 34, hlm. 97–113, Okt 2012, doi: 10.1016/j.knosys.2012.04.012.

[10] B. Billy, I. Kuswardayan, dan W. N. Khotimah, "Implementasi Artificial Intelligence pada game Defender of Metal City dengan menggunakan Finite State Machine," J. Tek. ITS, vol. 6, no. 2, hlm. A701-706, Okt 2017, doi: 10.12962/j23373539.v6i2.25151.

[11] M. H. Kurniawan dan D. Udjulawa, "Perbandingan Performa Algoritma Minimax Dan Alpha- Beta Pruning Pada Game Catur Cina," J. Algoritme, vol. 1, no. 1, hlm. 9, 2020.

[12] N. Azis, R. Puspasari, dan W. Verina, "Perancangan Aplikasi Chess Game Dengan Algoritma Alpha Beta Pruning," J. FTIK, vol. 1, no. 1, hlm. 325–331, 2020.

[13] N. P. Sijabat, M. A. Riad, J. S. Muda, dan D. B. Sanjaya, "Analisa Efektivitas Algoritma Minimax, Alpha Beta Pruning, dan Negamax dalam Penerapannya pada Permainan Papan (Board Game)," J. Ilmu Komput. Dan Sist. Inf. JIKOMSI, vol. 3, no. 2, hlm. 11, 2020.

[14] M. R. Handoko dan F. Indriani, "Perbandingan Algoritma Minimax Dengan Alphabeta Pruning Pada Permainan Mul- Mulan," Semin. Nas. Ilmu Komput. SOLITER, vol. 2, hlm. 7, 2019.

[15] J. N, V. K. B. C, M. S. Nadiger, N. R, dan S. H. N, "Alpha-Beta Pruning-A Streamline

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15863

Approach For Perceptive Game Playing," Int. Res. J. Mod. Eng. Technol. Sci., vol. 2, no. 6, hlm. 1306–1318, Jun 2020.

[16] D. S. Purnia, A. Rifai, dan S. Rahmatullah, "Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android," Semin. Nas. Sains Dan Teknol. Pros., hlm. 7, 2019.

[17] O. Alfina, "Sistem Informasi Mobile Assistant Mahasiswa Jurusan Sistem Informasi Fakultas Komputer Universitas Potensi Utama Berbasis Android," JITEKH, vol. 7, no. 01, hlm. 1–6, Mei 2019, doi: 10.35447/jitekh.v7i01.3.

[18] A. K. Effendi, R. Delima, dan A. R. Chrismanto, "Implementasi Algoritma Negascout Untuk Permainan Checkers," J. Inform., vol. 8, no. 1, Jul 2012, doi: 10.21460/inf.2012.81.112.

[19] A. Taqwiym, "Pendekatan Algoritma Greedy Untuk Menentukan Langkah Bidak Pada Permainan Checkers," Kurawal - J. Teknol. Inf. Dan Ind., vol. 1, no. 2, hlm. 89–97, Okt 2018, doi: 10.33479/kurawal.2018.1.2.89-97.

[20] E. R. Escandon dan J. Campion, "Minimax Checkers Playing GUI: A Foundation for AI Applications," dalam 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Agu 2018, hlm. 1–4, doi: 10.1109/INTERCON.2018.8526375.