IMPLEMENTASI ALGORITMA GREEDY UNTUK PENENTUAN LANGKAH PION PADA PERMAINAN LUDO

Primarani Dyah Bestari

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : kuyasipit@gmail.com

ABSTRAK

Ludo is a kind of board game played by two to four players take turns playing a pieces to reach the finish. In addition to playing with other players, the game can also be played with computers. For a computer to complete the game used greedy algorithm, there are four greedy algorithm to solve the ludo game Greedy by Hitter, No One Left Behind Greedy, Greedy by Quick Move, and Greedy by Blocking. These four algorithms will be combined to become a complete algorithm to determine the steps pieces to be played by computer.

In ludo game with a greedy algorithm to implement this using a waterfall methodology research software. Greedy algorithm with predefined scenarios dice, after it performed analysis and system design and analysis of algorithms and implementations. Applications created using Unity game then tested the algorithm implementation in order to know whether the implementation is done already as expected.

Testing is done by playing a game that has been built with the implementation of four algorithms which have been combined with algorithmic game that has not been combined. From the observation and testing found that playing on the fourth ludo greedy algorithm which has been incorporated, pieces reaches the finish faster.

Keyword : greedy algorithm, game, ludo

1. PRELIMINARY

Greedy algorithm is an algorithm that uses a type of problem-solving approach to find the maximum value while at every step. One of the problems that can be solved by the Greedy permaian Ludo. Ludo is derived from the Latin word "ludus", which means the game (games) [1]. Ludo game board game is a game type being played at least two people, players can fight against computer players or players. Ludo game can be solved by using a greedy algorithm, this algorithm is often used to solve the problem of finding solutions to get the results.

From previous research it is written that the four algorithms can be combined into one so as to produce a better algorithm in determining the pawn move. Therefore, in this study will be the fourth merger Greedy algorithms into one.

1.1 Greedy Algorithm

Greedy algorithm is an algorithm that uses a type of problem-solving approach to find the maximum value while at every step. While the maximum value is known as the local maximum. Most cases, the Greedy algorithm will not produce the most optimal solution, Greedy algorithms usually provide a solution that approaches the optimum value in quite right.

In many cases, the strategy Greedy algorithm does not produce an optimal solution, but Greedy may produce locally optimal solutions that approach the global optimal solution in a reasonable time. In language Greedy means "greedy". Greedy algorithms form the solution step by step. For each step, there are many options that need to be explored. Therefore, at each step must be made the best decision in determining the choice. Each step took the best option that can be obtained at the time without regard to future consequences (the principle of "take what you can get now!"), Hoping that by choosing a local optimum at each step will end up with a global optimum.

Four Greedy algorithms are implemented to determine the pawn steps, namely:

1 Greedy by Hitter

Pion players will move according to figures produced by the dice, if there is a player in front of the pawn pawn pawn opponent then the player will take the opponent pawn in front of her.

2 Greedy No One Left Behind

Pion players will move according to figures produced by the dice, if the pawn position players are in the rear, then pawn the player will move so as not to be in the rear.

3 Greedy by Quick Move

Pion players will move according to figures produced by the dice, if the pawn position players are close to the finish, then pawn the player will move to immediately reach the finish.

4 Greedy by Blocking

Pion players will move according to figures produced by the dice, if there is a player in deoan

pawn pawn pawn then the other players will step up and piled with pawn in front of her friends.

1.2 Ludo Game

Ludo is a kind of board game played by 2-4 players, where players compete to run their four pawns from start to finish based on the roll of the dice. Ludo is derived from the Latin word "ludus" which means game.



Figure 1 Board Games Ludo

Ludo board game or rectangular, in each corner there is a home base with four different colors, namely red, green, yellow, and blue, in Figure 2.1 are marked with the number 1 (one). Each home base has four pieces of the same color to the home base, respectively.

Pion runs clockwise rotating accordance, can be seen in Figure 2.1 the direction of the rate Ludo game marked by arrows. If the pawn has been spun as a purtaran then the pawn go into the lane in the middle according to the color of a home base and a pawn, respectively, as shown in figure 2.1 which is marked with the number 2 (two).

In Figure 2.1 are marked with the number 3 (three) is the finish or pawn goal after spinning and into the colored line in the middle.

1.3 Ludo Game Rules

Ludo game is played by two to four people. Each player gets four pieces with colors according to their own preferences, pawns are moved clockwise, turning to walk along the track is on board, to achieve the goal box in the middle of the board.

Each region or country has different rules depending ludo, but in this study the rules in the game of Ludo is as follows.

1 Players choose a color that will be played, then put the four pieces that match the color of the selected home base.

2 Players roll the dice in turn.

3 To remove the pawns into the path of the game, the player must earn six figures of dice being thrown.

4 Players move the pawn has been on the game in accordance with the numbers generated dice.

5. If players earn six figures again, then the player can choose to move the pawns are on track or remove pieces that are in a home base.

6 When the number appears players can melempat six dice again, and there is no limit if the dice are thrown generate consecutive sixes.

7 If the player stops at a pawn that has filled the opponents pawn pawns belonging to the opponents die and go back to home base belonging to the opponent.

8 If the player stops at a pawn filled with self-owned pawn, pawn tersebt then be stacked to make forts and opponents pawns can not pass any numbers generated by the dice.

9. Pion already passed one round into the corresponding colored lines with color pawns.

10 Pion will not move if the resulting figure is not appropriate dice to get into the finish. Example: pawn only need three more steps toward the finish, but the resulting number is four dice, then the pawn will be three steps forward and one step backward. And so on until the pawn gets in the finish.

1.4 Research Methods

Software development methods used in this study is the Waterfall method possible by reference Pressman. Broadly speaking, the Waterfall method includes several processes, which are:

1. requirement Analysis

This stage of analysis methods, algorithms and the needed resources to the development of the system. The analysis is carried out among other games analysis, feedback analysis, analysis algorithms will be implemented on the system, the analysis of non-functional requirements which include hardware and software, as well as functional analysis covering usecase diagrams, class diagrams, activity diagrams, and sequence diagrams .

2. system Design

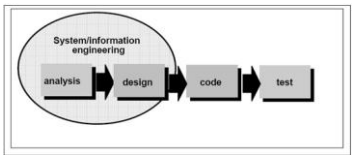
This phase is done before coding. This stage aims to provide an overview of what should be done and how it looks, like a storyboard design and system interface design.

3. Implementation

This stage of the programming is done on the system dibungun, in this stage of the algorithm implementation on a system built. This stage also conducted an examination of the system are made, whether it meets the desired function or not.

4. Integration and Testing

At this stage of testing to determine whether the system has been made in accordance with the design and there are errors or not.



Gambar 2 Waterfall Pressman[3]

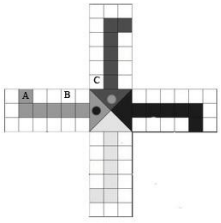
2. CONTENTS RESEARCH

Problems are found when the analysis was done in previous research that there are four written Greedy algorithms are used. In this study, four pawns step in determining the algorithm will be combined into one.

2.1 Analysis of Feedback

Greedy algorithms will be applied to the determination of the case analysis pawn move, the Ludo game is to search for a pawn is a priority for the step being examined.

Under conditions of a pawn that has some possibility to move, the Greedy algorithm is suitable to find the best solution. Every pawn is on the path of the game will see the status of pawns Which will be the top priority for the step.



Gambar 1 Feedback Analysis

A dan C adalah pion yang berwarna sama, sedangkan B adalah pion lawan dengan warna yang berbeda. Untuk penerapan algoritma *Greedy*, A dapat melangkah memakan B atau bertumpuk dengan C, kemungkinan lain adalah A melangkah mendahului C atau C melangkah semakin jauh mendekati *finish*.

2.2 Analisis Algoritma

Greedy by Hitter

Berikut ini adalah langkah-langkah pergerakan pion dalam menjalankan algoritma Greedy by Hitter :

1. Sistem mengecek giliran pion mana yang akan bermain
2. Pion hitam mendapatkan giliran bermain
3. Pion hitam mengocok dadu dan menghasilkan angka tiga
4. Sistem mengecek pion hitam dengan menggunakan perulangan apakah ada pion hitam yang dapat digerakkan dan terdapat pion lawan (pion merah) pada kotak jalur tujuan pion hitam
5. Jika ya maka pion hitam tersebut menjadi prioritas untuk melakukan algoritma Greedy by Hitter
6. Pion hitam maju tiga langkah lalu memakan pion merah.

Greedy No One Left Behind

Berikut ini adalah langkah-langkah pergerakan pion dalam menjalankan algoritma Greedy No One Left Behind :

1. Sistem mengecek giliran pion mana yang akan bermain
2. Pion hitam mendapatkan giliran bermain
3. Pion hitam mengocok dadu dan menghasilkan angka lima
4. Sistem mengecek pion hitam dengan menggunakan perulangan apakah ada pion yang berada paling belakang atau tidak
5. Jika ya maka pion tersebut menjadi prioritas untuk melakukan algoritma Greedy No One Left Behind
6. Pion hitam maju lima langkah sesuai dengan angka yang dihasilkan oleh dadu.

Greedy by Quick Move

Berikut ini adalah langkah-langkah pergerakan pion dalam menjalankan algoritma Greedy by Quick Move :

1. Sistem mengecek giliran pion mana yang akan bermain
2. Pion hitam mendapatkan giliran bermain
3 .Pion hitam mengocok dadu dan menghasilkan angka enam
4. Sistem mengecek pion hitam dengan menggunakan perulangan apakah ada pion yang berada dekat dengan finish atau tidak
5. Jika ya maka pion tersebut menjadi prioritas untuk melakukan Greedy by Quick Move
6. Pion hitam maju enam langkah sesuai dengan angka yang dihasilkan oleh dadu.

Greedy by Blocking

Berikut ini adalah langkah-langkah pergerakan pion dalam menjalankan algoritma Greedy by Blocking :

1. Sistem mengecek giliran pion mana yang akan bermain
2. Pion hitam mendapatkan giliran bermain
3. Pion hitam mengocok dadu dan menghasilkan angka empat
4. Sistem mengecek pion hitam dengan menggunakan perulangan apakah pada kotak jalur tujuan terdapat pion hitam lainnya atau tidak
5. Jika yam aka pion hitam tersebut menjadi prioritas untuk melakukan Greedy by Blocking
6. Pion hitam maju empat langkah lalu bertumpuk dengan pion hitam lainnyauntuk menghalangi jalan pion lawan.

Berikut ini adalah penjelasan dari pengimplementasian keempat algoritma Greedy pada permainan Ludo.

1. Himpunan kandidat = C, Himpunan kandidat (C) adalah pion-pion yang akan dimainkan. Terdapat empat warna pion masing-masing memiliki empat buah pion untuk dimainkan, total pion dalam permainan adalah enam belas.

C={1, 2. 3, 4, 5, 6, 7, 8, 9, 10 ,11 ,12 ,13 ,14 ,15 ,16}

Keterangan:

C={1, 2, 3, 4} adalah pion berwarna merah
C={5, 6, 7, 8} adalah pion berwarna biru
C={9, 10, 11, 12} adalah pion berwana hijau
C={13, 14, 15, 16} adalah pion berwana kuning

2. Himpunan solusi=S

Himpunan solusi (S) adalah kotak jalur yang akan dilalui oleh pion.

banyaknya kotak jalur yang akan dilalui pion selama permainan adalah lima puluh enam kotak kotak jalur.

S={f1 ,f2 ,f3 ,f4 ,f5 ,.., f56}

Keterangan:

F1 hingga f56 adalah kotak kotak jalur yang akan dilalui pion dari awal hingga finish.

3. Fungsi seleksi

Mengambil sebuah kandidat dari Himpunan Kandidat (C).

4. Fungsi kelayakan

Memeriksa apakah ada pion berdasarkan kandidat yang dipilih ada di dalam kotak jalur. Dilakukan dengan cara sistem melakukan perulangan apakah ada pion yang layak untuk digerakkan atau tidak. Jika ada maka pion tersebut akan bergerak, jika tidak maka sistem akan mencari kandidat lain untuk digerakkan.

5. Fungsi solusi

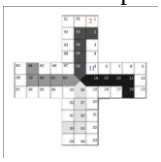
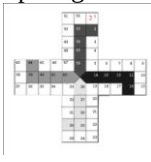
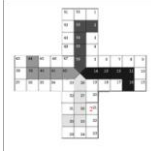
Jika angka yang dihasilkan dadu sama dengan kotak jalur tujuan pion, maka pion tersebut akan bergerak sesua angka yang dihasilkan oleh dadu.

6. Fungsi obyektif

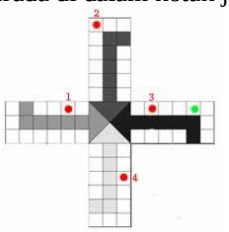
Jika kotak jalur yang dituju terdapat pion lawan maka dilakukanlah Greedy by Hitter, jika kotak jalur yang dituju kosong namun pion berada paling belakang maka dilakukanlah Greedy No One Left Behind, jika kotak jalur yang dituju kosong namu pion berada dekat dengan finish maka dilakukanlah Greedy by Quick Move, jika kotak jalur yang dituju terdapat pion yang berwarna sama maka dilakukanlah Greedy by Blocking.

Berapapun angka yang dihasilkan oleh dadu membuka kemungkinan

2.3 Contoh Kasus

3. Untuk *Greedy by Hitter* pion 2 akan maju empat langkah dan memakan pion 11 yang berada di depannya.
4. S= 
5. Untuk *Greedy No One Left Behind* pion 2 akan maju empat langkah agar tidak berara di paling belakang.
6. S= 
7. Untuk *Greedy by Quick Move* pion 2 akan maju empat langkah untuk mencapai *finish*.
8. S= 
9. Untuk *Greedy by Blocking* pion 2 akan maju empat langkah dan bergabung dengan pion 3 yang berada di depannya.
10. S= 

Berikut ini adalah contoh kasus jika di dalam kotak jalur terdapat empat pion sekaligus. Diasumsikan dadu menghasilkan angka tiga dan semua pion merah berada di dalam kotak jalur.



Gambar 3.4 Contoh Kasus 1

Sistem akan mengecek pion mulai dari yang paling dekat dengan *home base* secara berurutan, pada gambar 3.4 ditandai dengan pion merah bernomor satu. Sistem mengecek pion merah satu apakah pion tersebut berada dekat dengan *home base*, atau apakah ada pion lawan di depannya, atau apakah ada pion kawan di depannya, atau apakah pion tersebut berada dekat dengan finish. Pengecekan berulang hingga pion merah empat, setelah sistem selesai mengecek semua pion maka didapatkan pion merah tiga yang menjadi prioritas untuk melangkah dan melakukan *Greedy by Hitter*. Pion merah satu tidak melangkah karena pion tersebut berada dekat dengan *home base* namun angka yang dihasilkan oleh dadu tidak sesuai dengan *Greedy No One Left Behind* dimana pion yang paling belakang akan melangkah apabila dadu menghasilkan angka lima atau enam. Pion merah dua tidak melangkah karena pion tersebut berada jauh dari *home base* atau *finish* dan tidak ada pion kawan ataupun lawan di depannya. Pion merah empat tidak melangkah karena pion tersebut sama dengan pion merah dua meskipun pion berada paling depan namun untuk melakukan *Greedy by Quick Move* pion harus berada minimal sepuluh kotak kotak jalur dari kotak pertama kotak jalur aman.

2.2 Spesifikasi Kebutuhan Perangkat Lunak

Analisis perangkat lunak untuk mengimplementasikan algoritma Greedy pada penelitian ini akan dibagi kedalam dua bagian, yaitu analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional.

2.3 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan sistem yang menitik beratkan pada property prilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, kebutuhan perangkat lunak, serta kebutuhan user sebagai bahan analisis kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

1.Analisis Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi.

Table 1 Analisis Kebutuhan Perangkat Keras

Komputer	Spesifikasi Perangkat Keras

Prosesor	AMD E-350 1.60 GHz
Memory	4 GB
Resolusi layar	1366 x 178 pixel
Harddisk	320 GB
Keyboard dan mouse	-

2.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan.

2.2.1 Usecase

Usecase diagram merupakan gambaran fungsional dari interaksi antara actor dengan sistem



Gambar 2 Usecase

Definisi use case berfungsi untuk memperjelas proses yang terdapat pada setiap use case.

Table 2 Definisi Use Case

No	*Use Case*	Deskripsi
1	Masuk ke menu utama	Proses dimana pemain masuk ke dalam menu utama permainan
2	Memilih warna dan jumlah pemain	Proses pemilihan pion dan jumlah pemain yang akan bermain
3	Start game	Proses measuk ke dalam area permainan
4	Memainkan permainan	Proses memainkan permainan
5	Pause	Proses menghentikan permainan sementara
6	Melakukan *Greedy by Hitter*	Proses menjalankan greedy by hitter
7	Melakukan *Greedy No One Left Behind*	Proses menjalankan greedy no one left behind
8	Melakukan *Greedy by Quick Move*	Proses menjalankan greedy by Quick move
9	Melakukan *Greedy by Blocking*	Proses menjalankan greedy by blocking
10	Keluar permainan	Keluar dari permainan

Tabel 3 Skenario Usecase Pause

Identifikasi	
Nama	Pause
Tujuan	Menghentikan permainan sementara
Deskripsi	Proses menghentikan permainan sementara
Aktor	Pemain
Kondisi Awal	Pemain berada di area permainan
Kondisi Akhir	Pemain berada di area permainan Pemain keluar permainan
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1.Menekan tombol "esc" pada keyboard	
	2.Menampilkan menu pesan "continue" dan "quit"
3a.Memilih menu pesan "continue"	
3b.Memilih menu pesan "quit"	
	4a.kembali menjalankan permainan
	4b.menampilkan menu pesan "yes" dan "no"
5a.memilih menu pesan "no"	
5b.memilih menu pesan "yes"	
	6a.kembali menampilkan menu pesan "continue" dan "quit"
	6b.keluar dari permainan

2.2.2 Class Diagram

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem.



Gambar 3 Class Diagram

2.3 Skenario Pengujian

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi permainan Ludo. Skenario pengujian dari aplikasi permainan Ludo dapat dilihat pada tabel 4.2.

Table 4.2 Skenario Pengujian

No	Komponen yang Dijuji	Butir Uji	Jenis Pengujian
1	Menu Utama	Masuk ke menu utama	*Blackbox*
2	Memilih warna dan pemain	Memilih tombol dadu	*Blackbox*
		Memilih pemain "CPU"	*Blackbox*
		Mematikan pemain	*Blackbox*
3	Main Permainan	Memilih tombol *start*	*Blackbox*
		Memilih tombol dadu	*Blackbox*
		Memilih pion	*Blackbox*
4	Pause permainan	Memilih tombol continue	*Blackbox*
		Memilih tombol quit	*Blackbox*
		Memilih tombol yes	*Blackbox*
		Memilih tombol no	*Blackbox*
5	Algoritma *Greedy*	Greedy by Hitter	*Whitebox*
		Greedy No One Left Behind	*Whitebox*
		Greedy by Quick Move	*Whitebox*
		Greedy by Blocking	*Whitebox*

1. Blackbox

Tabel 1 Pengujian Blackbox

Kasus Dan Hasil Uji			
Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Tekan tombol "esc" pada keyboard	Sistem melakukan proses untuk mem-*pause* permainan dan menampilkan pesan "quit" dan "continue"	Sistem menampilkan pesan "quit" dan "continue"	[ ] Tidak berhasil
Klik tombol "continue"	Sistem melakukan proses untuk menampilkan papan permainan	Sistem menampilkan papan permainan	[√] Berhasil [ ] Tidak berhasil
Klik tombol "quit"	Sistem melakukan proses untuk menampilkan pesan "yes" dan "no"	Sistem menampilkan pesan "yes" dan "no"	[√] Berhasil [ ] Tidak berhasil
Klik tombol "no"	Sistem melakukan proses untuk menampilkan kembali pesan "quit" dan "continue"	Sistem menampilkan kembali pesan "quit" dan "continue"	[√] Berhasil [ ] Tidak berhasil
Klik tombol "yes"	Sistem melakukan proses untuk menutup aplikasi	Sistem menutup aplikasi	[√] Berhasil [ ] Tidak berhasil

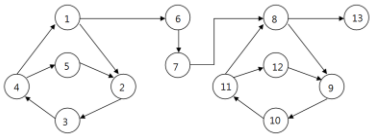
2. Whitebox

Bagian ini merupakan kasus algoritma *Greedy* untuk menguji aplikasi yang sudah dibangun dengan menggunakan metode *Whitebox*, berdasarkan skenario pengujian aplikasi yang sudah dijabarkan. Pengujian algoritma *Greedy* merupakan proses dimana akan diuji terhadap pion-pion dalam pemilihan langkah pada permainan Ludo. Berikut ini adalah algoritma *Greedy* yang diimplementasikan pada permainan Ludo dengan menggunakan bahasa pemrograman *JavaScript.*

```
    function EatPiece(a:int,b:int){
        var ts : String;
1       switch(a){
2         case 1:
              ts = "P1";
              break;
3         case 2:
              ts = "P2";
```

```
        break;
4       case 3:
        ts = "P3";
        break;
5       case 4:
        ts = "P4";
        break;
    }
    var gos : GameObject[];
    gos = GameObject.FindGameObjectsWithTag(ts);
6   for(var go in gos){
      var tv : int;
      tv = go.GetComponent.<piece1>().cField;
7     if(tv == b){
        go.GetComponent.<piece1>().cField = 0;
        go.GetComponent.<piece1>().road = 0;
        go.GetComponent.<piece1>().nearHome = false;
        go.GetComponent.<piece1>().pMode = PieceMode.base;
        var bp : int;
        var bs : String;
        var bpos : Vector3;
        bp = go.GetComponent.<piece1>().baseField;
8       switch(a){
9         case 1:
            bs = gameScript.p1base[bp];
            break;
10        case 2:
            bs = gameScript.p2base[bp];
            break;
11        case 3:
            bs = gameScript.p3base[bp];
            break;
12        case 4:
            bs = gameScript.p4base[bp];
            break;
        }
        bpos = GameObject.Find(bs).transform.position;
        go.transform.position = bpos;
        toEatP = 0;
        toEatF = 0;
        return;
      }
13  }
}
```

Flowgraph algoritma *Greedy by Blocking* dan *Greedy by Hitter*



Gambar 4. *Flowgraph Greedyby Hitter*

Keterangan :



= Menggambarkan kondisi

→ = Menggambarkan aksi

Cyclomatc Complexity V(G)

Dari gambar 4.10 dapat dihitung *cyclomatic complexity* sebagai berikut :

V(G) = E – N + 2

Di mana E adalah edge, dan N adalah node

V(G) = E – N + 2

= 16 – 13 + 2

= 5

3 PENUTUP

Berdasarkan hasil pengujian implementasi antarmuka yang dilakukan, algoritma *Greedy* yang diterapkan pada permainan Ludo telah sesuai dengan apa yang diharapkan, yaitu menentukan langkah pion apabila terjadi kondisi-kondisi tertentu. Begitu pula pengujian *whitebox* yang dilakukan, masing-masing *independent path* yang dihasilkan bisa dijalankan dan dieksekusi serta tidak ada perulangan yang tak berhingga

3.1 Kesimpulan

Berdasarkan hasil dari pengujian dan implementasi permainan Ludo, maka dapat diambil kesimpulan sebagai berikut. Penerapan algoritma Greedy pada permainan ini telah sesuai dengan apa yang diharapkan, langkah pion sesuai dengan algoritma yang diimplementasikan, yaitu algoritma Greedy yang telah digabungkan dapat menentukan langkah pion apabila terjadi kasus dengan kondisi tertentu.

3.2 Saran

Permainan Ludo ini masih dapat dikembangkan lebih lanjut dengan perkembangan spesifikasi kebutuhan pengguna sistem yang harus dipenuhi dalam mencapai tahap yang lebih tinggi dan kinerja sistem yang lebih baik. Algoritma yang diterapkan dapat dibuat lebih random dalam menentukan langkah pion.

DAFTAR PUSTAKA

[1] Lio Franklyn Kemit. Penentuan Langkah dengan Greedy dalam Permainan Ludo, 2012

[2] Dr. Juliansyah Noor, S.E., M.M., Metode Penelitian Skripsi, Thesis, Disertasi, dan Karya Ilmiah.: KENCANA PRENADA MEDIA GROUP, 2012

[3] Andi Setiya Wardana (2013, Oktober). Pengembangan Sistem Dengan Metode Waterfall [online]. Available : http://andisetiya.blog.widyatama.ac.id/2013/10/02/pengembangan-sistem-informasi-dengan-metode-waterfall/ (Diakses 8 Juni 2014 23:45)