

## RANCANG BANGUN GAME *TREASURE OF LABYRINTH* DENGAN ALGORITMA *BACKTRACKING* BERBASIS *ANDROID*

Rio Andriyat Krisdiawan<sup>1</sup>, Ramdoni<sup>2</sup>, Aji Permana<sup>3</sup>

Fakultas Ilmu Komputer Universitas Kuningan

Jalan Tjut Nyak Dhien No. 36 A Cijoho Kuningan Jawa Barat 45513 Telepon (0232) 2875097

[rioandriyat@uniku.ac.id](mailto:rioandriyat@uniku.ac.id) / [rioandriyat@gmail.com](mailto:rioandriyat@gmail.com), [2015081090@uniku.ac.id](mailto:2015081090@uniku.ac.id), [aji@uniku.ac.id](mailto:aji@uniku.ac.id)

*Game* merupakan salah satu media hiburan yang menjadi pilihan hampir setiap orang untuk menghilangkan kejenuhan ataupun hanya mengisi waktu luang, ada juga *game* dijadikan hobi dan terlebih lagi *game* saat ini dijadikan olahraga elektronik (*e-sport*). Saat ini *game* memiliki banyak jenis permainan, salah satunya adalah jenis *game* teka teki (*puzzle*). Salah satu *game* teka-teki adalah *game* labirin. Labirin merupakan sebuah *puzzle* yang memiliki bentuk percabangan yang kompleks dan memiliki banyak jalan buntu. Pada saat ini banyak *game* yang berjenis labirin, akan tetapi masih jarang *game* labirin yang dilengkapi karakter, musuh dan sebuah arena labirin yang dinamis atau bisa berubah apabila permainan diulangi. Untuk membuat *game* labirin yang memiliki arena labirin yang dinamis diperlukan sebuah pembangkit labirin atau bisa disebut *labyrinth generator* dengan memanfaatkan algoritma *backtracking*. Algoritma ini cukup mangkus untuk digunakan dalam beberapa penyelesaian masalah dan juga untuk memberikan kecerdasan buatan dalam *game*. Metodologi perangkat lunak yang digunakan dalam pembuatan *game* yaitu *GDLC (Game Development Life Cycle)*. *GDLC* adalah model pengembangan sebuah *game* yang menerapkan pendekatan iteratif yang terdiri dari 6 fase pengembangan, dimulai dari fase *innitiation*, *pre-production*, *production*, *testing*, *beta* dan *realese*. Untuk pembangkitan arena labirin menggunakan algoritma *Backtracking* sedangkan untuk pengujian *game* ini menggunakan *UAT (User Acceptment Test)*. Hasil penelitian ini berupa *game* petualangan berbasis teka-teki labirin yang diterapkan pada *mobile android*. *Game* ini dimainkan guna melatih pemain dalam pemecahan masalah dan sarana hiburan.

**Kata Kunci** : *Game Puzzle, GDLC (Game Development Life Cycle), Algoritma Backtracking, UAT, Android*

*Games are one of the entertainment media which becomes a choice of almost everyone to get rid of boredom or just fill in spare time, there are also games used as hobbies and moreover games are now used as electronic sports (e-sports). Currently the game has many types, one of them is a type of puzzle game (puzzle). One of puzzle games is a maze game. The labyrinth is a puzzle that has a complex branching form and has many dead ends. At the moment there are many labyrinth type games, but there are rarely labyrinth games equipped with characters, enemies and a dynamic maze that can change if the game is repeated. To create a labyrinth game that has a dynamic labyrinth arena requires a labyrinth generator or it can be called a labyrinth generator by utilizing the backtracking algorithm. This algorithm is powerful enough to be used in some problem solving and also to provide artificial intelligence in the game. The software methodology used in making games is the GDLC (Game Development Life Cycle). GDLC is a game development model that adopts an iterative approach consisting of 6 development phases, starting from the innitiation, pre-production, production, testing, beta and realese phases. For the generation of the labyrinth arena, it uses the Backtracking algorithm while for testing this game uses the UAT (User Acceptment Test). The results of this study in the form of map puzzle based adventure game e that is applied to mobile android. This game is played to train players in problem solving and entertainment facilities.*

**Keywords**: *Puzzle Game, GDLC (Game Development Life Cycle), Backtracking Algorithm, UAT, Android*

### 1. PENDAHULUAN

Perkembangan didunia *Game* saat ini mengalami perkembangan yang

sangat cepat terutama pada *Game Android*, dikarenakan perkembangan *Smartphone Android* saat ini semakin

canggih dan semakin banyak, seiring banyaknya pengguna *Smartphone Android* maka kebutuhan hiburan melalui media *Smartphone Android* semakin banyak terutama pada *Game Android*.

*Game* bergenre petualangan yang mengangkat unsur teka-teki yang bertemakan *labyrinth/maze* pada saat ini dikembangkan hanya sebatas menelusuri ruang atau jalan pada sebuah labirin untuk menemukan sebuah jalan keluar, tentu saja itu menjadi sebuah kekurangan karena hanya sebatas menelusuri sebuah labirin untuk menemukan jalan keluar saja, karena kurang adanya tantangan seperti adanya musuh pada permainan, contohnya seperti halnya pada permainan "*Mine Maze*" yang berbasis *Android*.

Dalam permainan "*Mine Maze*" tidak ada musuh dan tidak ada karakter jelas pada permainan tersebut, dan juga *labyrinth* pada game tersebut bersifat statis atau tidak berubah ketika permainan diulangi, sehingga ketika pemain memainkan permainan pada *level* yang sama maka permainan akan sangat mudah. Tentu saja hal ini mengurangi tantangan pada permainan tersebut, terlebih lagi tidak adanya musuh pada arena *labyrinth* pada permainan tersebut mengurangi tantangan dan unsur kesenangan pada game tersebut. Oleh karena itu penulis tertarik untuk mengembangkan *game* petualangan yang bertemakan *labyrinth* dengan penambahan karakter dan musuh beserta arena *labyrinth* yang dinamis yang selalu berubah ketika memulai permainan.

Untuk membangun sebuah *labyrinth* yang dinamis/dapat berubah setiap permainan dimulai tentu diperlukan sebuah sistem untuk membangkitkan sebuah *labyrinth* atau bisa disebut *labyrinth generator*. Salah satunya memanfaatkan algoritma

*backtracking*. Algoritma ini cukup mangkus untuk digunakan dalam beberapa penyelesaian masalah dan juga untuk memberikan kecerdasan buatan dalam game.

*Game* yang akan dibangun bertujuan mengumpulkan harta karun dan mengalahkan musuh didalam sebuah arena *labyrinth* serta mencari jalan keluar sebagai *goal* pada setiap *levelnya*. Dengan adanya arena *labyrinth* yang dinamis serta musuh dan harta karun pada arena *labyrinth* akan membuat permainan menjadi lebih seru. Harta karun yang akan dibangun pada permainan ini bertemakan ciri khas Kabupaten Kuningan Jawa Barat, seperti senjata kujang, bokor, ikan dewa, batik dan yang terakhir adalah maskot kuningan yaitu patung kuda.

Berdasarkan uraian latar belakang masalah diatas, maka penulis dapat mengidentifikasi masalah yang perlu diatasi sebagai berikut:

1. Belum adanya *Game* yang berjudul *Treasure of labyrinth* berbasis android.
2. Belum adanya *Game Treasure of labyrinth* yang dilengkapi karakter, musuh dan harta karun bertemakan kearifan local kabupaten kuningan.
3. Tidak adanya karakter, musuh dan harta karun pada *Game Mine Maze* yang sudah ada di *playstore*.
4. Tidak adanya *labyrinth generator* pada *Game Mine Maze*.

Dalam melakukan penelitian ini penulis membatasi permasalahan agar aplikasi *game* yang dibangun sesuai dengan rencana dan spesifikasi yang ditentukan antara lain:

1. *Game treasure of labyrinth* yang dibangun dimainkan di sistem operasi *android* dan dapat dimainkan pada *android* minimal versi *android kitkat*.
2. *Game treasure of labyrinth* hanya dapat dimainkan oleh *sigle player*.

3. *Game treasure of labyrinth* ini hanya memiliki 5 tingkat kesulitan (*Level*).
  - a. *Level 01*, pemain disajikan sebuah *labyrinth* berukuran *grig* 5x5 dan musuh sebanyak 5, coin, kunci sebagai pembuka pintu pada jalan keluar sebagi *goal* dari permainan, *item heal* untuk menambah darah dari pemain dan harta karun.
  - b. *Level 02*, pemain disajikan sebuah *labyrinth* berukuran *grig* 7x7 dan musuh sebanyak 10, coin, kunci sebagai pembuka pintu pada jalan keluar sebagi *goal* dari permainan, *item heal* untuk menambah darah dari pemain dan harta karun.
  - c. *Level 03*, pemain disajikan sebuah *labyrinth* berukuran *grig* 9x9 dan musuh sebanyak 15, coin, kunci sebagai pembuka pintu pada jalan keluar sebagi *goal* dari permainan, *item heal* untuk menambah darah dari pemain dan harta karun.
  - d. *Level 04*, pemain disajikan sebuah *labyrinth* berukuran *grig* 11x11 dan musuh sebanyak 20, coin, kunci sebagai pembuka pintu pada jalan keluar sebagi *goal* dari permainan, *item heal* untuk menambah darah dari pemain dan harta karun.
  - e. *Level 05*, pemain disajikan sebuah *labyrinth* berukuran *grig* 13x13 dan musuh sebanyak 25, coin, kunci sebagai pembuka pintu pada jalan keluar sebagi *goal* dari permainan, *item heal* untuk menambah darah dari pemain dan harta karun
4. *Game engine* yang digunakan untuk membangun aplikasi *game treasure of labyrinth* adlah *UNITY*.
5. Bahasa pemograman yang digunakan untuk membangun *game treasure of labyrinth* adalah *C#*.
6. Algoritma yang digunakan adalah algoritma *Bacaktracking* yang

digunakan sebagai *labyrinth generator*.

Adapun tujuan dari dilakukanya penelitian ini adalah sebagai berikut :

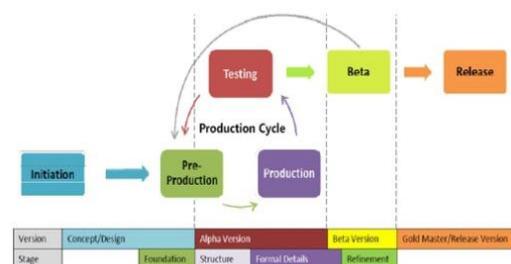
1. Untuk membangun *game treasure of labyrinth* berbasis *android*.
2. Untuk mengimplementasikan algoritma *Backtracking* pada *game treasure of labyrinth* sebagai *labyrinth generator*.

Manfaat dari dilakukannya penelitian ini adalah untuk memahami dan menerapkan model pengembangan *GDLC* dan *algoritma Backtracking* dalam mengembangkan *game treasure of labyrinth*.

## 2. METODELOGI PENELITIAN

### 2.1. Metodologi Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Game Development Life Cycle* (*GDLC*). *GDLC* adalah suatu proses pengembangan sebuah game yang menerapkan pendekatan iteratif yang terdiri dari 6 fase pengembangan, dimulai dari fase *innitiation*, *pre-production*, *production*, *testing*, *beta* dan *realese*.



**Gambar 1.** Fase dan Proses *GDLC*  
Krisdiawan, R. A. (2018). Implementasi Model Pengembangan Sistem *GDLC* dan Algoritma Linear Congruential Generator pada Game Puzzle. *NUANSA INFORMATIKA*, 12(2).

Pada gambar fase dan proses *GDLC* menjelaskan bahwa *GDLC* memiliki 6 fase yang dapat dilakukan secara iteratif, yaitu tahapan *Initiation*, *Ppre-production*, *Testing*, *Beta* serta *Realease* :

1. *Initiation* (Inisiasi)  
Penulis membuat sebuah konsep kasar dari *game treasure of labyrinth* yaitu berupa story berupa berpetualang disebuah labirin mencari jalan keluar serta mengumpulkan harta karun dan mengalahkan musuh yang ada pada arena labirin. Target pemain *game* yang penulis bangun adalah semua umur.
2. *Pre-Production* (Pra-Produksi)  
Penulis membuat desain *game* berfokus pada mendefinisikan *genre* permainan, *gameplay*, *game* mekanik, alur cerita, tantangan dan faktor kesenangan
3. *Production* (Produksi)  
Penulis membuat aset – aset *game* yang akan dibangun seperti *Charachter*, *Enemy*, *Item*, *Treasure*, *Desain UI*, *Music*, *Sound Effect*, dan lainnya. Penulis juga memprogram *game* dan menyelesaikan *game* tersebut.  
Asset Game :  
Karakter : Karakter manusia dengan senjata. Hanya satu karakter dengan 3 kostum.  
Enemy : Bermacam macam mosters  
Item : Senjata, darah, kunci,  
Harta karun : Koin emas, Harta karun kuningan.
4. *Testing* (Pengujian)  
Penulis menguji fungsi fungsi serta fitur yang dibangun pada *game* apakah berfungsi dengan baik atau tidak, dengan menggunakan metode pengujian *White Box* dan *Black Box*. Ketika penulis mendapatkan *bug* atau kesalahan pada sistem penulis mendokumentasikan dan

menganalisis *bug* tersebut dan memperbaikinya serta dilakukan analisis apakah sudah layak untuk masuk ke tahap selanjutnya yaitu *Beta*.

#### 5. Pengujian *Beta*

Penulis melakukan pengujian yang melibatkan pihak eksternal untuk melakukan pengujian sistem yang disebut sebagai *tester*. Pengujian ini memiliki 2 jenis pengujian yaitu pengujian *beta* tertutup dan pengujian *beta* terbuka. Disini penulis menggunakan jenis pengujian *beta* tertutup, yaitu penulis melakukan pengujian dengan cara mengundang pihak individu yang memungkinkan untuk menjadi penguji. Tujuannya untuk mencoba sistem dan mencari *bug* yang terdapat pada sistem, jika penguji mendapatkan *bug* pada sistem maka akan dilakukan dokumentasi menggunakan metode UAT (*User Accepment Test*) untuk mendapatkan *feddback* dari penguji guna untuk menilai apakah *game* yang dibangun sudah layak untuk melanjutkan ke tahap selanjutnya yaitu *realese*.

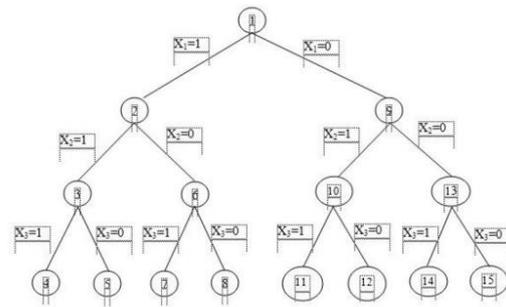
#### 6. *Realese* (Rilis)

Pengembangan sistem sudah mencapai tahap akhir dan siap untuk diliris ke publik. Untuk tahap ini penulis baru melakukan publish dengan menshare *game* tersebut secara itranet.

## 2.2. Metode Pemecahan Masalah dengan Algoritma Backtracking

Algoritma runut balik (*Backtracking*) pertama kali diperkenalkan oleh D.H Lehmer pada tahun 1950. Algoritma ini cukup mangkus untuk digunakan dalam beberapa penyelesaian masalah dan juga untuk memberikan kecerdasan buatan dalam *game*. Beberapa *game* populer semisal Sudoku, Labirin,

Catur juga bisa diimplementasikan dengan menggunakan algoritma runut balik. Algoritma runut balik berbasis pada DFS (Depth First Search) sehingga aturan pencariannya akan mengikuti kepada aturan pencarian DFS yaitu dengan mencari solusi dari akar ke daun (dalam pohon ruang solusi) dengan pencarian mendalam. Simpul-simpul yang sudah dilahirkan (diperiksa) dinamakan simpul hidup (live node). Simpul hidup yang sedang diperluas dinamakan simpul-E atau Expand Node.



**Gambar 2.** Pohon Solusi (Bambang Fahrudin 2016).

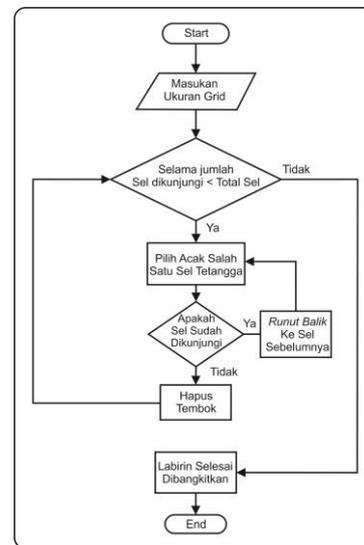
Untuk *flowchart* proses pembangkitan arena labirin di lihat di gambar 3.

Untuk menerapkan metode runut-balik, properti berikut didefinisikan:

1. Solusi persoalan. Solusi dinyatakan sebagai vektor n-tuple:  $X=(x_1, x_2, \dots, x_n)$ ,  $x_i$  anggota himpunan berhingga  $S_i$  Mungkin saja  $S_1 = S_2 = \dots = S_n$ . Contoh:  $S_i = \{0,1\}$   $S_i = 0$  atau  $1$ . Fungsi pembangkit nilai  $x_k$  Dinyatakan sebagai:  $T(k)$

$T(k)$  membangkitkan nilai untuk  $x_k$ , yang merupakan komponen vektor solusi.

Fungsi Pembatas (fungsi kriteria) Dinyatakan sebagai:  $B(x_1, x_2, \dots, x_k)$  Fungsi pembatas menentukan apakah  $(x_1, x_2, \dots, x_k)$  mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk  $x_{k+1}$  dilanjutkan, tetapi jika tidak, maka  $(x_1, x_2, \dots, x_k)$  dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

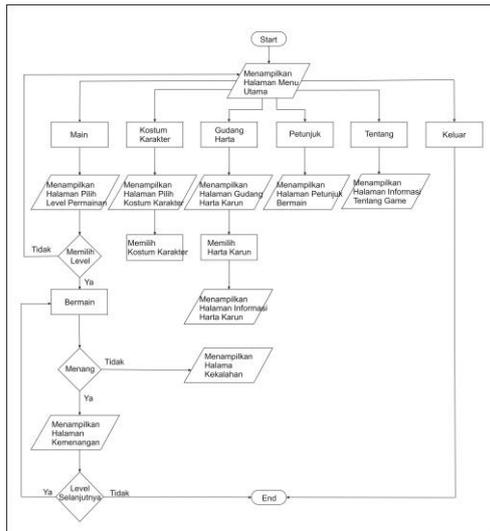


**Gambar 3.** Flowchart Sistem Proses Pembangkitan Labirin.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Tahap Pre-Production (Pra-Produksi)

##### 3.1.1. Sistem game yang diusulkan



Gambar 4. Diagram Flow Menu Game

### 3.1.2. Story Board

Tabel 1. Story Board Scene 1

Scene 1	Memasuki area labirin	SScene 1/5
Lokasi	Didalam labirin	
Aksi	Saat permainan dimulai, player akan memasuki arena labirin	

Tabel 2. Story Board Scene 2

Scene 2	Mencari peti harta karun	SScene 2/5
Lokasi	Didalam labirin	
Aksi	Berjalan menelusuri labirin dan menemukan peti harta karun	

Tabel 3. Story Board Scene 3

Scene 3	Bertarung dengan enemy	SScene 3/5
---------	------------------------	------------

Lokasi	Didalam labirin
Aksi	Melanjutkan perjalanan dan bertemu dengan enemy lalu player menembaknya untuk mengalahkannya

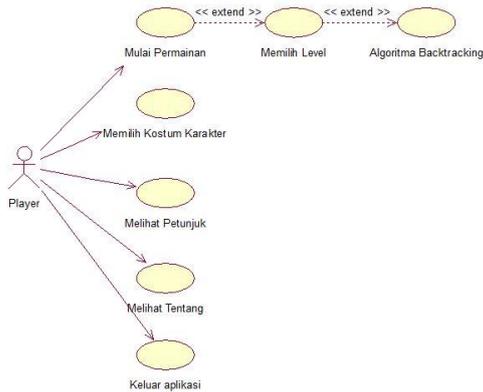
Tabel 4. Story Board Scene 4

Scene 4	Mendapatkan kunci pintu goal	SScene 4/5
Lokasi	Didalam labirin	
Aksi	Melanjutkan perjalanan dan menemukan kunci pintu goal	

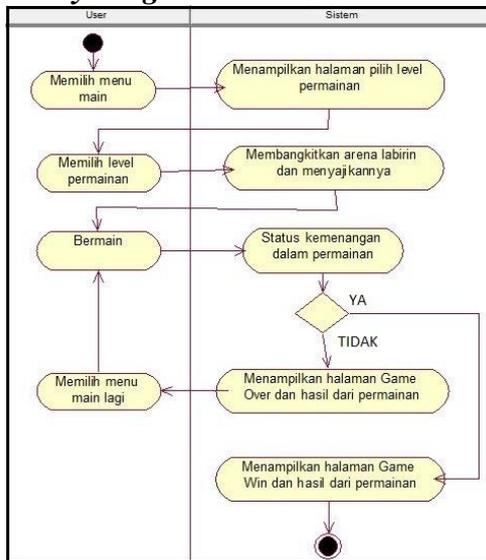
Tabel 5. Story Board Scene 5

Scene 5	Permainan selesai	SScene 5/5
Lokasi	Didalam labirin	
Aksi	Player melanjutkan perjalanan, dan dengan didapatkannya kunci pintu goal, player dapat menyelesaikan permainan dengan memasuki pintu goal.	

### 3.1.3. Perancangan Sistem Game Use Case Diagram

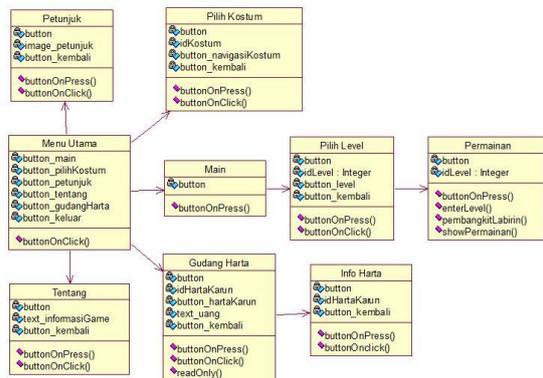


**Gambar 5. Use Case Diagram Sistem Game Treasure Of Labirin. Activity Diagram Main**

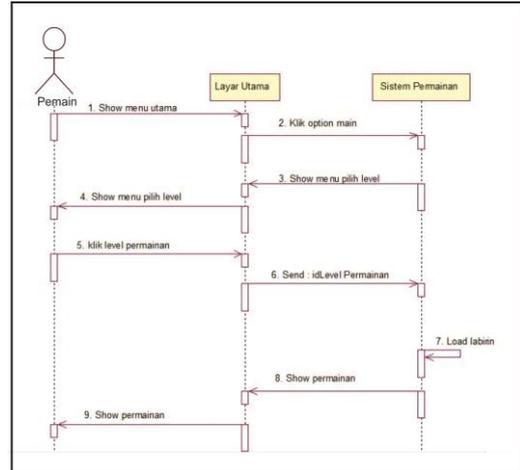


**Gambar 6. Activity Diagram Main.**

**Class Diagram**



**Gambar 7. Class Diagram Game Treasure Of Labyrinth. Sequence Diagram**



**Gambar 8. Sequence Diagram Main 3.2. Tahap Production Halaman Manu Utama**

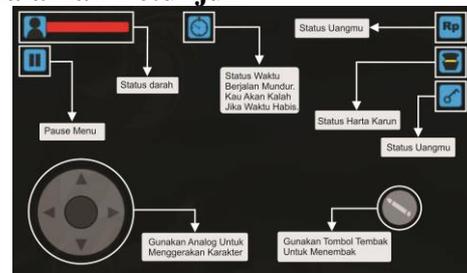


**Gambar 9. Interface Halaman Utama Halaman Pilih Level**



**Gambar 10. Interface Halaman Pilih Level**

**Halaman Petunjuk**



**Gambar 11. Interface Halaman Petunjuk**

### Halaman Kostum Karakter



Gambar 12. Interface Pilih Kostum Karakter

### Halaman Gudang Harta



Gambar 13. Interface Gudang Harta Halaman Info Harta Karun



Gambar 14. Interface Halaman Info Harta Karun

### Halaman Permainan



Gambar 15. Interface Halaman Permainan

### 3.3. Tahap Testing

#### Pengujian Black Box

Tabel 6. Pengujian Black Box Halaman Menu Utama

Pengujian Black Box dari halaman menu utama				
Tombol	Aksi Pemain	Reksi Sistem		Status

Menu main	Pemain menekan tombol main	Sistem menampilkan halaman pilih level	Sesuai harapan	Valid
Menu Petunjuk	Pemain menekan tombol petunjuk	Sistem menampilkan halaman menu petunjuk	Sesuai harapan	Valid
Menu Kostum Karakter	Pemain menekan menu kostum karakter	Sistem menampilkan halaman menu pilih kostum karakter	Sesuai harapan	Valid
Menu Gudang Harta	Pemain menekan tombol gudang harta	Sistem menampilkan halaman menu gudang harta	Sesuai harapan	Valid
Menu Tentang	Pemain menekan tombol tentang	Sistem menampilkan halaman tentang aplikasi game	Sesuai harapan	Valid
Menu Keluar	Pemain menekan tombol keluar	Aplikasi berhenti	Sesuai harapan	Valid

### 4. KESIMPULAN

Dari penelitian yang telah dilakukan bahwa dengan menggunakan algoritma *Backtracking* pada *game treasure of labyrinth* untuk membangkitkan sebuah *labyrinth* yang dinamis, maka dapat disimpulkan bahwa:

1. Dengan memanfaatkan algoritma *backtracking* penulis dapat membangun sistem pembangkit labirin dinamis.
2. Model pengembangan *GDLC* sangat cocok dalam membantu dalam membuat sebuah game.
3. Dengan dimasukkannya icon-icon khas kabupaten kuningin sebagai harta karun dapat memperkenalkan sesuatu ciri khas dari kabupaten kuningin jawa barat.

## 5. SARAN

*Game* yang penulis kembangkan masih mempunyai beberapa kekurangan dan perlu adanya pengembangan kedepannya, maka dari itu penulis memiliki harapan dan saran dari hasil temuan peneliti diantaranya

1. Perlu adanya penambahan *level* dan karakter pada *game* sehingga permainan pada *game* akan lebih menantang.
2. Perlu adanya pengembangan *game* secara online untuk membuat sebuah permainan yang dapat dimainkan lebih dari 1 player disaat yang sama sehingga permainan dapat lebih menarik.

## DAFTAR PUSTAKA

- [1]. Andi Juansyah, (2015). "Pembangunan Aplikasi Child Tracker Berbasis Assisted-Global Positioning System (A-GPS) Dengan Platform Android", Bandung : Jurnal Ilmiah Komputer dan Informaika (KOMPUTA).
- [2]. Bambang Fahrudi, (2016). "Penerapan Algoritma Backtracking Pada Permainan Capsa Banting", Medan : Jurnal Riset Komputer (JURIKOM).
- [3]. Gellysa Urva dan Helmi Fauzi S., (2015). "Pemodelan UML E-Marketing Minyak Goreng", Sumatra Utara : Jurnal Teknologi dan Sistem Informasi.
- [4]. Gun Gun M., (2017). "Pembelajaran Dasar Algoritma Dan Pemograman Menggunakan El-Goritma Berbasis Web", Bandung : Jurnal Teknik Mesin (JTM).
- [5]. Hans Alfon E., Imam Kuswardayan dan Dr. Eng nanik S., (2016). "Rancang Bangun Game Berhitung Spaceship dengan Pengendalian Suara Menggunakan Speech Recognition Plugin pada Unity", Surabaya : Jurnal Teknik (ITS).
- [6]. Jenifer Rori, Steven Sentinuwo dan Stanley Karouw, (2016). "Perancangan Aplikasi Panduan Belajar Pengenalan Ortodonsia Menggunakan Animasi 3D", Manado : E-Jurnal Teknik Informatika).
- [7]. Lewis, W. E., (2009). Software Testing and Continuous Quality Improvement. ketiga ed. Boca Raton: CRC Press.
- [8]. Nidhra, Srinivas dan Jagruthi Dondeti. (2012). Black box and White box Testing Techniques – A Literature Review. Internasional Journal of Embedded System and Applications (IJESA) Vol.2, No.2.
- [9]. Krisdiawan, R. A. (2018). Implementasi Model Pengembangan Sistem GDLC dan Algoritma Linear Congruential Generator pada Game Puzzle. *NUANSA INFORMATIKA*, 12(2).
- [10]. Rizky, Soetam. (2011). "Konsep Dasar Rekayasa Perangkat Lunak", Jakarta : Prestasi Pustaka, 2011.
- [11]. Roger S. Pressman, (2002). Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu), ANDI Yogyakarta.
- [12]. Suhendar, Andrianto, dan Gunadi, H. (2002). Visual Modelling menggunakan UML dan Rational Rose, Informatika Bandung, Bandung.
- [13]. Wahyu Pratama, (2014). "Game Adventure Misteri Kotak Pandora", Purwokerto : Jurnal Telematika.
- [14]. Wamiliana, Dian Kurniasari dan Dolly Yudhidtira, (2013). "Implementasi Aalgortima Backtrack untuk Pencarian Solusi Knight's Tour Problem pada Papan Catur", Lampung : Ilmu

Komputer Unila Publishing  
Network.