Journal of Information System, Informatics and Computing

Vol.3 No.2 Desember 2019
e-ISSN : 2597-3673 *(Online)*
p-ISSN : 2579-5201 *(Printed)*

RANCANG BANGUN *GAME "LEGENDS OF SPACESHIP"* MENGGUNAKAN *GAME* MAKER STUDIO

Rahmat Tirta Kusuma[1], Astriana Mulyani [2], Harsih Rianto
Program Studi Teknik Informatika[1,2], Program Studi Sistem Informasi[3]
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri Jakarta[1,2]
Universitas Bina Sarana Informatika[3]
rahmatti2111@nusamandiri.ac.id[1] , astriana.atm@nusamandiri.ac.id[2], harsih.hhr@bsi.ac.id[3]

Abstrak

Semua orang pasti pernah mengenal apa itu *Game*, mulai dari anak – anak hingga orang tua masa kini pasti pernah meminkannya. *Game* adalah salah satu jenis aktivitas bermain, yang didalamnya dilakukan dalam konteks berpura – pura namun terlihat seperti realitas. *Game* juga dapat diartikan sebagai kegiatan penyelesaian masalah, didekati dengan sikap yang menyenangkan, *Game* yang bagus adalah yang bisa membuat pemain berpartisipasi secara aktif, bisa menigkatkan dan melatih kelincahan pemain dan mempunyai tantangan yang tepat, tidak terlalu sedikit atau terlalu banyak. Penulis memilih latar ruang angkasa yang penulis anggap cukup menarik untuk menjadi *Game* petualang menggunakan *Game* Maker Studio. Metode penelitian yang digunakan terbagi menjadi beberapa tahap yaitu pengumpulan data, perancangan, pembuatan, dan tahap uji coba. Proses pembuatan *Game* ini menggunakan bahasa pemrograman *Game* Maker Language dan software Adobe Photoshop 7.0 untuk pembuatan desain objek dan user interface dan beberapa software pendukung lainnya. Dapat ditarik kesimpulan, bahwa telah berhasil dibuat *Game* ini dengan genre Action yang terbagi menjadi 3 mode permainan yang berbeda, dan *Game* ini dapat menjadi hiburan, meningkatkan ketangkasan dan kelincahan dan menjadi sarana penghilang jenuh setelah beraktifitas seharian.

Kata Kunci: *Game, Game Maker Studio, Agility, Spaceship*

I. PENDAHULUAN

Game merupakan kata dalam bahasa inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks yang tidak serius atau dengan tujuan untuk refreshing. Game juga bersifat *adiktif* atau dapat menimbulkan ketergantungan bagi pemakainya. Game juga akan sangat berguna jika dimanfaatkan secara positif[1].

Semua orang pasti pernah mengenal apa itu *Game*, mulai dari anak–anak hingga orang tua masa kini pasti pernah memainkan suatu video *Game* dalam hidupnya. *Game* adalah salah satu jenis aktivitas bermain, yang di dalamnya dilakukan dalam konteks berpura-pura namun terlihat seperti realitas, dimana pemainnya memiliki tujuan untuk mendapatkan suatu kemenangan serta dilakukan sesuai dengan aturan permainan yang dibuat[2].

Game yang bagus adalah yang bisa membuat pemain berpartisipasi secara aktif, bisa meningkatkan *ability* atau kelincahan pemain dan mempunyai tantangan yang tepat, tidak terlalu sedikit atau terlalu banyak. *Ability* (kemampuan, kecakapan, ketangkasan, bakat dan kesanggupan) merupakan tenaga (daya kekuatan) untuk melakukan suatu perbuatan bisa merupakan kesanggupan bawaan sejak lahir, atau merupakan hasil latihan atau hasil praktek [3].

Selain menjadi media hiburan, *Game* juga dapat menjadi sebuah media untuk melatih daya motorik seseorang karena *Game* mengandung tantangan-tantangan yang beragam. Ada berbagai jenis *game* yang berkembang, baik *game* edukasi, *game* petualangan, *game* strategi, *game* kartu, dan masih banyak jenis *game* lainnya. Dari sekian banyak jenis *game* yang ada, salah satunya adalah jenis *game* petualangan[4]. Dengan bermain *game* berjenis petualangan diharapkan pemainnya dapat meningkatkan kelincahan dalam berfikir dan mengambil keputsan secara cepat dan tepat.

Pada penelitian ini penulis menggunakan algoritma *collision detection*. Algoritma ini melakukan proses pengecekan apakah beberapa buah objek spasial saling bertumpuk atau tidak. Jika ternyata ada paling sedikit dua buah objek yang bertumpuk, maka kedua objek tersebut dikatakan saling bertumpukkan[4]. Pada ruang spasial dua dimensi. Objek yang bertumpuk berarti objek spasialnya beririsan.

II. LITERATUR DAN METODE

1. Game

Journal of Information System, Informatics and Computing
Vol.3 No.2 Desember 2019
e-ISSN : 2597-3673 *(Online)*
p-ISSN : 2579-5201 *(Printed)*

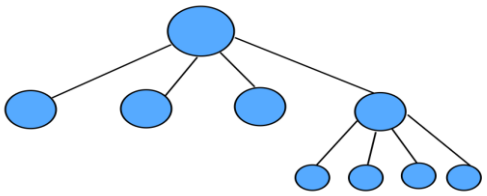
Game adalah salah satu jenis aktivitas bermain, yang di dalamnya dilakukan dalam konteks berpura-pura namun terlihat seperti realitas[2]. Dimana pemainnya memiliki tujuan untuk mendapatkan satu kemenangan serta dilakukan dengan sesuai aturan permainan yang dibuat. Macam-macam jenis game antara lain:

a. Aksi (*Action*)
   Game jenis aksi adalah game yang paling populer. Game berjenis aksi ini membutuhkan kemampuan reflek pemainnya. Pemain yang menjalankan game ini seolah-olah berada dalam suasan yang terdapat dalam permainan. Pemain diberikan keleluasaan untuk membangun proyek tertentu dengan bahan baku yang sudah dipersiakan sesuai setting game yang dijalankan[5].
b. *Role Playing Game* (RPG)
   Dalam permainan game RPG pemain dapat memilih satu karakter untuk dimainkan. Seiring dengan naiknya level game, karakter tersebut dapat berubah, bertambah kemampuan, bertambah senjata atau bertambah hewan peliharaan yang terdapat dalam permainan.
c. Strategi
   Dalam game jenis strategi menitikberatkan pada kemampuan berfikir dan berorganisasi. Game strategi dibedakan menjadi dua jenis, yaitu Turn Based Strategy dan Real Time Strategi. Turn Based Strategy memberikan kesempatan pada pemain untuk menjalankan taktiknya. Disaat pemain mengambil langkah bermain, pihak lawan menunggu untuk dan sebaliknya. Sedangkan Real Time Strategy mengharuskan pemain membuat keputusan dan secara bersamaan pihak lawan untuk membuat keputusan bersamaan.
d. Balapan (*Racing*)
   Pemain dapat memilih kendaraan, lalu melaju di arena balap yang tedapat pada tampilan layar game. Tujuannya permainan ini adalah untuk mencapai garis finish tercepat.
e. Olah Raga
   Game jenis ini membawa olahraga ke dalam sebuah komputer atau konsol. Biasanya *gameplay* dibuat semirip mungkin dengan kondisi olahraga yang sebenarnya. Sehingga pemain dapa menjalankan permainan seperti sedang malakukan olah raga.
f. Puzzel
   Game jenis puzzle menyajikan teka-teki, menyamakan warna bola, perhitungan matematika, menyusun balok, atau mengenal huruf dan gambar. Sehingga pemain yang menyelesaikan game ini akan medapatkan pengalaman sesuai dengan puzzle yang dijalankan.
g. Permainan Kata
   Game jenis ini sering dirancang untuk menguji kemampuan pemainnya dengan bahasa atau untuk mengeksplorasi sifat- sifatnya. Permainan kata umumnya digunakan sebagai sumber hiburan, namun juga bisa digunakan untuk tujuan proses pendidikan.

2. Algoritma *Collision Detection*

Menurut setiyawan dan winarno algoritma *collisions detections* untuk mendeteksi tabrakan antar objek sehingga objek bisa saling bereaksi dan tidak hanya saling menembus saat objek saling bersentuhan atau saling bertumpuk, metode ini juga memiliki fungsi untuk memeriksa apakah dua buah objek spasial saling menumpuk dan bersentuhan[4].

Menurut Tukan, algoritma *collision detection* adalah proses pendeteksian tabrakan antara dua objek. Sebenarnya dalam simulasi penelitian ini tabrakan tidak hanya terjadi antara dua objek, tetapi dapat juga terjadi antara satu objek dengan banyak objek sehingga dibutuhkan collision detection yang akurat[6]. Collision detection juga berguna untuk menentukan posisi dari satu objek dengan objek yang lain agar tidak ada objek yang saling menembus, sehingga simulasi yang akan



Gambar 1. *Model Algoritma Collision Detection*

dibuat memiliki kesamaan dengan realita yang ada. Gambar 1 mempelihatkan model algoritma *collision detection.*

3. Game Maker

Game Maker merupakan bahasa khusus yang dikembangkan pada tool game maker. Penggunaannya adalah dengan menggunakan blok-blok ikon yang operasikan dengan cara klik drag and drop[7]. Game Maker Language dibuat oleh Mark Overmars, pembuat Game Maker sendiri.

Game Maker merupakan game engine yang dapat digunakan dalam proses pembuatan sebuah game dengan antarmuka yang mudah digunakan oleh pengguna. Terdapat 10 macam asset yang harus diperhatikan oleh pembuat game yaitu:

a. Sprite

Journal of Information System, Informatics and Computing
Vol.3 No.2 Desember 2019
e-ISSN : 2597-3673 *(Online)*
p-ISSN : 2579-5201 *(Printed)*

Sprite adalah representasi visual dari semua benda dalam game. Sprite dapat terdiri dari satu gambar saja atau lebih dari satu gambar sehingga terlihat seperti gerak animasi.

b. *Sounds*
Dengan sounds kita dapat menyertakan suara latar atau efek suara sehingga game terasa lebih hidup. Dalam *Game Maker* sudah terdapat beberapa efek suara yang telah disediakan

c. *Backgrounds*
Adalah gambar besar yang digunakan untuk latar belakang sebuah game yang berlangsung, bisa dimodifikasi latar belakang dengan aplikasi flash sehingga terlihat bergerak.

d. *Path*
Sebagai *waypoint* atau lintasan yang diikuti oleh suatu objek dalam game. Dengan path kita dapat menentukan kearah mana sebuah objek bergerak seperti pergerakan musuh dalam game

e. *Scripts*
Tempat untuk coding dengan bahasa pemrograman yang disebut *Game Maker Language* dapat dipakai jika diperlukan.

f. *Fonts*
Fonts digunakan untuk menambahkan font yang akan di gunakan dalam game. Font biasanya digunakan dengan fungsi text drawing.

g. *Time Lines*
Time lines merupakan fungsi waktu yang diberikan dalam sebuah object. Timeline ini memberikan waktu pada suatu objek dalam melakukan sebuah event atau action pada game dengan waktu yang sudah ditentukan

h. *Objects*
Objects merupakan benda hidup dari sebuah sprite yang akan bertindak dalam permainan. Sebuah objek diberikan pergerakan melalui event yang harus direaksi pada saat game ini dioperasikan, dan bagaimana sebuah obyek harus bereaksi ketika ada suatau aksi.

i. *Room*
Room adalah ruang untuk game berjalan dan juga sebagai tempat untuk objek ditempatkan untuk bergerak, kita dapat membuat banyak rooms sehingga terdapat rintangan yang berbeda – beda pada setiap game pada masing-masing room, room dapat dihubungkan ke room lain sehingga suatu objek dapat berpindah-pindah

j. *Event*
Event adalah pergerakan yang dilakukan suatu objek apabila terjadi suatu peristiwa (*event*) pada objek tersebut, misalnya benturan, input keyboard, pergerakan mouse, dan lain-lain.

4. *Unified Modeling Language* (UML)

Menurut Haviludin dalam Prasetya mengemukakan bahwa UML adalah bahasa standar untuk membuat rancangan perangkat lunak. UML biasanya digunakan untuk menggambarkan, menentukan, menyusun dan mendokumentasikan artefak dari software-intensive system[8].

Menurut Rahadi, Satoto, & Windasari menerangkan bahwa UML digunakan untuk melakukan permodelan perangkat lunak dengan menggunakan tools yang ada. Dengan permodelan menggunakan UML, rekayasa dan pengembangan perangkat lunak dapat dilakukan dengan fokus pengembangan dan desain perangkat lunak[2].

UML terdiri dari banyak diagram antara lain:

a. *Use Case Diagram*
Menurut Singkoh, Lumenta, & Tulenan menerangkan bahwa *Use Case Diagram* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Use Case mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Use Case Diagram menampilkan aktor mana yang menggunakan use case mana, use case mana yang memasukkan use case lain dan hubungan antara aktor dan use case[1].

b. *Activity Diagram*
Activity Diagram digunakan untuk menampilkan rangkaian kegiatan, menunjukkan alur kerja dari suatu titik awal ke titik akhir keputusan, merinci banyak jalur yang ada dalam perkembangan peristiwa yang terkandung dalam kegiatan[9].

c. *Class Diagram*
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua

Journal of Information System, Informatics and Computing
Vol.3 No.2 Desember 2019
e-ISSN : 2597-3673 *(Online)*
p-ISSN : 2579-5201 *(Printed)*

kelas yang membentuk arsitektur sistem yang dibuat[10].

d. Sequence Diagram

Sequence Diagram adalah diagram interaksi yang disusun berdasarkan urutan waktu. Setiap diagram sekuensial mempresentasikan satu flow dari beberapa flow di dalam use case, definisi lain dari sequence diagram ialah merupakan Interaction Diagram yang digunakan untuk menjelaskan eksekusi sebuah skenario semantik. Sequence Diagram juga digunakan untuk menjelaskan interaksi antar objek dalam urutan waktu[9].

e. Deployment Diagram

Deployment diagram digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis Object-Oriented yang akan dibangun[11].

5. Pengujian

Pada pengujian aplikasi penulis menggunakan 3 metode penggujian yang digunakan yaitu:

a. White Box Testing

Pengujian yang didasarkan pada detail prosedur dan alur logika kode program. Pada kegiatan whitebox testing, tester melihat source code program dan menemukan bugs dari kode program yang diuji. Intinya whitebox testing adalah pengujian yang dilakukan sampai kepada detail pengecekan kode program[12].

b. *Black Box Testing*

Black Box Testing atau Pengujian Kotak Hitam atau juga disebut Behavioral Testing, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik Black Box Testing memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program[8].

c. Pengujian *Beta*

Beta testing adalah pengujian oleh pemakai di lingkungan operasi pemakai. Evaluasi beta testing dilakukan oleh pengguna. Mereka diberitahukan prosedur evaluasi, diamati proses penggunaannya, diwawancarai lalu dinilai dan dilakukan revisi[13].

III. METODE PENELITIAN

Model Pengembangan yang digunakan dalam metode penelitian ini ialah model waterfall. Pengembangan sistem dikerjakan secara terurut mulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung.

1. Analisa Kebutuhan Sistem

Merupakan tahapan awal dimana dilakukan identifikasi masalah, usulan pemecahan masalah dan analisa kebutuhan system yang difokuskan untuk pembuatan piranti perangkat lunak. System yang digunakan untuk merancang *Game Lagends of Spaceship* meliputi:

a. Dibuatkan tampilan animasi dan suara dengan kualitas yang baik agar lebih menarik para pemain.
b. Dibuatkan lavel-lavel permainan dari yang paling mudah sampai paling sulit.
c. Statistik hasil dari permainan yang sudah dilakukan.

2. *Design* (Perencanaan)

Pada tahap selanjutnya dilakukan pembuatan model dari perangkat lunak. Maksud pembuatan model ini adalah untuk memperoleh pengertian yang baik terhadap aliran data dan control, proses-proses fungsional, tingkah laku operasi dan informasi-informasi yang terkandung didalamnya. Terdiri dari aktivitas utama pemodelan proses, pemodelan data dan desain antarmuka.

3. *Code Generation* (Pengkodean)

Pada tahapan pengkodean yaitu melakukan penerapan hasil rancangan ke dalam bentuk yang dapat dibaca dan dimengerti oleh komputer. Pada tahapan ini hasil dari perancangan mulai diterjemahkan kedalam Bahasa mesin melalui Bahasa pemrograman. Adapun jenis Bahasa pemrograman yang digunakan oleh penulis adalah Game Maker Language menggunakan game engine Game Maker Studio varsi 1.4.1

4. Testing (Pengujian)

Testing adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi desain dan pengkodean. Pada testing ini penulis menggunakan pengujian 3 tahapan pengujian yaitu: *white box, black box* dan pengujian Beta

5. *Support* (Pemeliharaan)

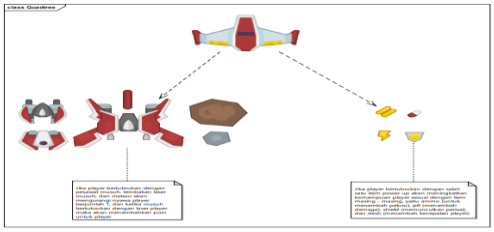
Pada tahap ini, merupakan tahap pemeliharaan atau maintenance terhadap aplikasi yang ada. Untuk mendukung desain dan perancangan sebuah sistem yang memadai diperlukan hardware maupun software yang sesuai dengan kebutuhan diantaranya operation system, software design, dan perangkat yang menunjang.

IV. HASIL DAN PEMBAHASAN

1. Perancangan Objek

Journal of Information System, Informatics and Computing
Vol.3 No.2 Desember 2019
e-ISSN : 2597-3673 *(Online)*
p-ISSN : 2579-5201 *(Printed)*

Pada Game ini diterapkan algoritma bernama Quadtree. Algoritma Quadtree adalah algoritma yang digunakan pada Game untuk pengecekan collision (benturan/tabrakan) dua objek berbeda pada arena permainan dua dimensi secara efisien.

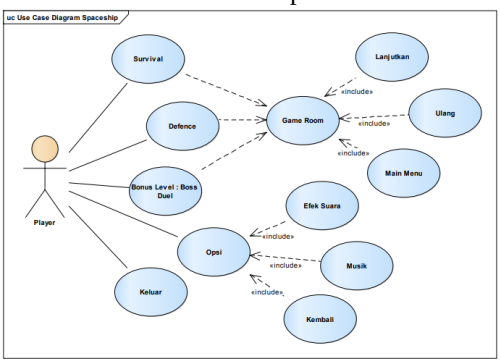


Gambar 2. *Skema collision pada Game*

Dalam Game ini algoritma quadtree diterapkan pada objek player sebagai objek utamanya. Jika objek player ini berbenturan dengan objek lain akan mengakibatkan suatu kejadian atau kondisi berbeda dari setiap benturan yang terjadi dengan objek lain. Gambar 2 merupakan hasil skema perancangan objek menggunakan algoritma *collision Detection.*

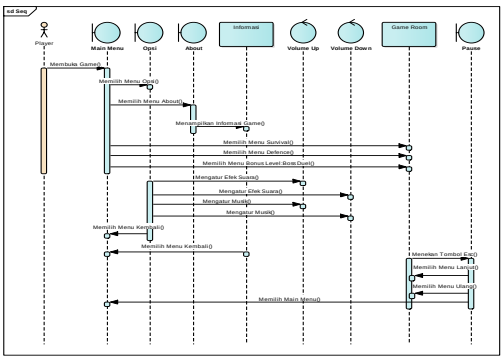
2. Permodelan UML

Gambar 3 adalah use case diagram dari rancangan game yang dibuat oleh peneliti. Dalam use case diagram ini terdapat aktifitas yang akan dilakukan oleh pemain game pada saat bermain. Seperti pemilihan level permainan, pemilihan bonus permainan, opsi settingan permainan dan tombol keluar dari permainan.



Gambar 3. *Use Case Diagram*

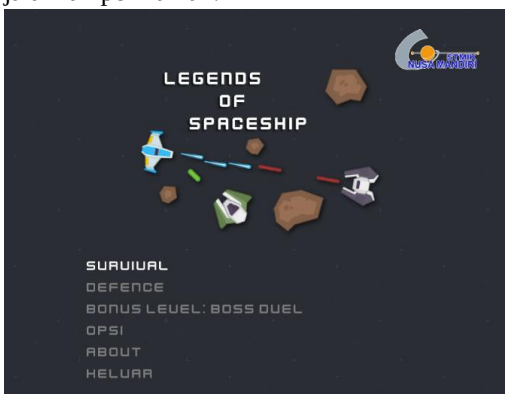
Gambar 4 adalah Squence Diagram yang dibuat untuk rancangan program game yang dibuat. Squence Diagram berfungi untuk mengontrol progam yang dibuat sudah sesuai dengan desain yang sudah direncanakan.



Gambar 4. *Sequence Diagram*

3. Implementasi *Interface*

Gambar 5 merupakan interface halaman utama dari Game yang dibuat. Dalam menu utama ini ditampikan menu-menu yang dapat dipilih oleh pemain sebelum menjalankan permainan.



Gambar 5. Menu Utama

Gambar 6 adalah tampilan pada saat game dijalankan. Pada tampilan saat permainan dijalankan maka akan ada objek yang akan menyerang objek utama.

Gambar 6.Tampilan saat *Game* dijalankan

Journal of Information System, Informatics and Computing
Vol.3 No.2 Desember 2019
e-ISSN : 2597-3673 *(Online)*
p-ISSN : 2579-5201 *(Printed)*

Gambar 7 adalah tampilan about dari progam game yang dibuat. Pada tampilan about dimunculkan informasi bagaimana permainan game dapat dijalankan.

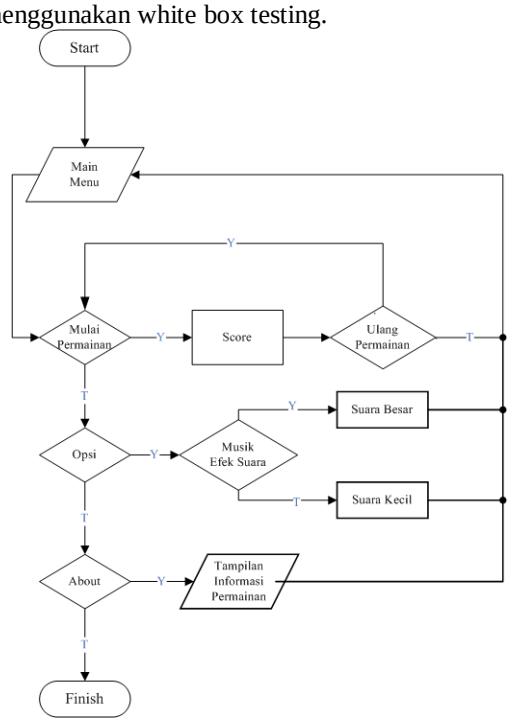
Gambar 7. About

4. Pengujian

Pada tahapan pengujian penulis menggunakan 3 model penggujian yang dilakukan pada proses pembuatan program game yaitu:

a. *White Box Testing*

Metode pengujian *white box* menggunakan *flowchart* atau bagan alir. Diagram alir atau *flowchart* merupakan bagian yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Fungsinya adalah memberikan gambaran secara garis besar untuk program atau aplikasi yang dibuat[14]. Gambar 8 adalah flowchart yang dibuat untuk proses penggujian menggunakan white box testing.



Gambar 8. Flowchart White Box Testing

b. *Black Box Testing*

Metode pengujian yang dilakukan selanjutnya ialah *black box testing*. *Black Box Testing* atau pengujian kotak hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak, artinya teknik *Black Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. Tabel 1 adalah daftar hasil dari pengujian black box testing yang sudah dilakukan.

c. Pengujian *Beta*

Beta Testing adalah pengujian oleh pemakai di lingkungan operasi pemakai, evaluasi *beta testing* dilakukan oleh pengguna, mereka diberitahukan prosedur evaluasi, diamati proses penggunaannya,

Journal of Information System, Informatics and Computing
Vol.3 No.2 Desember 2019
e-ISSN : 2597-3673 *(Online)*
p-ISSN : 2579-5201 *(Printed)*

diwawancarai lalu dinilai dan dilakukan revisi. Pengujian ini terdiri dari 8 pertanyaan yang disebarkan oleh 10 responden dan menggunakan skala *guttman* untuk perhitungannya.

Tabel 1. Hasil Pengujian Black Box

Skenario Uji	*Test Case*	Hasil yang diharapkan	Ket
Tampil menu utama	Menuju ke menu utama	Tampil halaman menu utama	*Valid*
Menampilkan menu Opsi	Memilih menu Opsi	Tampil halaman menu Opsi	*Valid*
Mengatur besar *volume* suara	Masuk ke menu Opsi dan mengatur *volume* musik dan efek suara	*Volume* suara musik dan efek suara bisa ditambah dan dikurangi	*Valid*
Menampilkan menu *About*	Memilih menu *About*	Tampil halaman menu *About*	*Valid*
Menampilkan *pause menu* di *Game room* ketika sedang dalam permainan	Menekan tombol Esc	*Game Pause* dan menampilkan menu *pause*	*Valid*
Menampilkan menu *Game Over* ketika pemain kehabisan nyawa di permainan	Membenturkan pemain dengan musuh hingga habis nyawa	*Game* berhenti dan menampilkan menu *Game Over*	*Valid*
Menampilkan menu *You Win!* Ketika pemain menyelesaikan misi permainan dengan baik	Menyelesaikan misi di permainan dengan baik	*Game* berhenti dan menampilkan menu *You Win!*	*Valid*
Keluar dari *Game*	Memilih menu Keluar	*Game* berakhir dan menutup	*Valid*

Tabel 2. Skor Skala Guttman

Kriteria	Nilai/Skor
Ya	1
Tidak	0

Tabel 3 adalah hasil kuesioner yang didapat dari 10 responden terhadap 8 pertanyaan beserta persentase dari masing-masing pertanyaan.

Tabel 3. Hasil Kuesioner

Pertanyaan	Jawab	Jml	%
1. Apakah aplikasi ini sangat mudah digunakan?	Ya	10	100
	Tidak	0	
2. Apakah *Game* ini dapat menghilangkan kejenuhan setelah beraktifitas?	Ya	6	60
	Tidak	4	
3. Apakah aplikasi ini merangsang ketangkasan dan kelincahan dalam memainkannya?	Ya	7	70
	Tidak	3	
4. Apakah tampilan grafis *Game* ini menarik?	Ya	5	50
	Tidak	5	
5. Apakah tingkat kesulitan *Game* ini sangat besar?	Ya	2	20
	Tidak	8	
6. Apakah suara efek dan *background* di *Game* ini terdengan jelas?	Ya	9	90
	Tidak	1	
7. Apakah anda pernah bermain *Game* seperti ini sebelumnya?	Ya	6	60
	Tidak	4	
8. Apakah anda senang memainkan *Game* ini?	Ya	10	100
	Tidak	0	

Dari data diatas diperoleh jumlah skor dan seluruh pertanyaan yang diajukan dan diperoleh 55 jawaban "Ya" dan 25 jawaban "Tidak". Sesuai dengan intepretasi skor yang telah ditentukan sebelumnya, maka jumlah skor hasil pengumpulan data angket dari seluruh responden harus dikonversikan kedalam bentuk presentase dengan menggunakan rumus berikut:

$$Dp = \frac{n}{N}\ x\ 100\% \qquad Dp = \frac{n}{N}\ x\ 100\%$$

$$Dp = \frac{55}{80}\ x\ 100\% \qquad Dp = \frac{25}{80}\ x\ 100\%$$

$$Dp = 68{,}75\% \qquad Dp = 31{,}25\%$$

Berdasarkan perhitungan diatas respon terhadap Game ini didapat nilai sebesar 68,75% yang menjawab "Ya" dan berada dalam persentase 50%-100% yang dapat dikategorikan "Baik". Kesimpulan yang diperoleh bahwa tampilan aplikasi, visualisasi, manfaat, serta kemudahan menggunakan aplikasi sudah baik, sehingga Game ini mampu menghilangkan kejenuhan dan bisa melatih ketangkasan dan kelincahan pemain (*agility*).

V. KESIMPULAN

Game yang dirancang dan dibangun sudah dapat memenuhi kebutuhan untuk menghibur, meningkatkan *ability* atau ketangkasan dan kelincahan pemain dan sebagai sarana mengatasi kejenuhan setelah beraktifitas. Game ini hanya bisa di mainkan dengan komputer dengan sistem operasi Windows, menggunakan keyboard dan mouse untuk memainkannya. Game ini menggunakan game engine Game Maker Studio v1.4.1 dari YoYo Games. Game ini terdiri dari beberapa mode yang memiliki tingkat kesulitan yang berbeda-beda di setiap mode Dari hasil pengujian yang telah dilakukan diketahui bahwa kompleksitas siklomasinya valid dan fungsi-fungsi yang ada berjalan dengan benar, tidak ada cacat dan sebagaimana mestinya sesuai harapan penulis.

REFERENASI

[1]R. T. Singkoh, A. S. M. Lumenta, and V. Tulenan, "Perancangan Game FPS (First Person Shooter) Police Personal Training," *E-Journal Tek. Elektro Dan Komput.*, vol. 5, no. 1, pp. 28–34, 2016.

[2]M. R. Rahadi, K. I. Satoto, and I. P. Windasari, "Perancangan Game Math Adventure Sebagai Media Pembelajaran Matematika Berbasis Android," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 1, p. 44, 2016.

[3]E. Ekojono, D. A. Irawati, L. Affandi, and A. N. Rahmanto, "Penerapan Algoritma Fisher-Yates pada Pengacakan Soal Game Aritmatika," in *SENTIA*, 2017, vol. 9, pp. 101–106.

[4]D. Setiyawan and E. Winarno, "Game Petualangan Si Toole Untuk Mempromosikan Wisata Kabupaten Grobogan Menggunakan Metode Collision Detection," in *Prosiding SINTAK*, 2018, no. 2012, pp. 318–324.

[5] W. A. Hamka and A. Gani, "Rancang Bangun Game Edukasi Berbasis Web Dan Android Menggunakan Adobe Flash Cs5 Dan Action Script 3.0," *IJIS - Indones. J. Inf. Syst.*, vol. 1, no. 2, pp. 78–88, 2016.

[6] E. A. Tukan, "Penerapan Augmented Reality pada Game Book," in *Seminar Nasional Teknologi Informasi dan Multimedia*, 2015, pp. 6–8.

[7] K. T. Martono, "Pengembangan game dengan menggunakan Game Engine Game Maker," *J. Sist. Komput.*, vol. 5, no. 1, pp. 23–30, 2015.

[8] A. Prasetya, M. Ugiarto, and N. Puspitasari, "Pengembangan Aplikasi Game Shoot'em Up Star Assault Dengan Game Maker Studio," in *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*, 2017, vol. 2, no. 1, pp. 4–7.

[9]U. Rusmawan, *Teknik Penulisan Tugas Akhir dan Skripsi Pemograman*. 2019.

[10]M. Ropianto, "Pemahaman Penggunaan Unified Modelling Language," *JT-IBSI*, vol. 1, no. 1, pp. 43–50, 2016.

[11]M. I. Martin, "Apa yang dimaksud dengan Deployment Diagram," *https://www.dictio.id/t/apa-yang-dimaksud-dengan-deployment-diagram/15125*, 2018. .

[12]Binus University, "Perbedaan White Box Testing dan Black Box Testing," *Binus University*, 2016. [Online]. Available: http://scdc.binus.ac.id/himsisfo/2016/10/perbedaan-white-box-testing-dan-black-box-testing/.

[13]A. B. Mutiara, R. Awaludin, A. Muslim, and T. Oswari, "TESTING IMPLEMENTASI WEBSITE REKAM MEDIS ELEKTRONIK OPELTGUNASYS DENGAN METODE ACCEPTANCE TESTING," in *Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2014)*, 2014, vol. 8, no. Kommit 2014, pp. 1–7.

[14]F. Marzian and M. Qamal, "Game RPG ' The Royal Sword ' Berbasis Desktop Dengan Menggunakan Metode Finite State Machine (FSM)," *Sist. Inf.*, pp. 61–96, 2017.