

MEMBANGUN *SIDE SCROLLING GAME* “GRIM REAPER FLYING IN THE DARKNESS”

Siti Lailiyah¹⁾, Reza Andrea²⁾, Asbudi³⁾

^{1,2,3}Teknik Informatika, STMIK Widya Cipta Dharma

^{1,2,3}Jl. M. Yamin No.25, Samarinda, 75123

E-mail : lail59a@gmail.com¹⁾, reza@bibirdesign.com²⁾, dhaycops@gmail.com³⁾

ABSTRAK

Game Grim Reaper Flying In The Darkness adalah sebuah *game* bergenre *Side Scrolling Game* dimana dalam permainannya mengisahkan sebuah karakter yang bernama *grim reaper* menghancurkan musuh. Dengan *Genre Side Scrolling Game* ialah *genre* yang sangat cocok untuk dimainkan untuk umur 6 tahun ke atas karena kontrol *gamenya* yang tidak rumit dan dapat membantu untuk melatih konsentrasi.

Adapun tahap tahap dalam membangun *game Grim Reaper Flying In The Darkness* ialah dengan menggunakan tahapan pengembangan multimedia yaitu *concept, material collecting, assembly, testing, Distribution*, agar dalam membangun *game grim reaper flying in the darkness* dengan terstruktur. Sehingga dibutuhkan algoritma *collision detection* untuk mendeteksi objek satu dengan objek lainnya. Dimana dengan menggunakan algoritma *collision detection* diharapkan menghasilkan sebuah permainan yang dapat digunakan sebagai sarana hiburan.

Kata Kunci: *Swishmax, Grim Reaper Flying In The Darkness, Side Scrolling Game.*

1. PENDAHULUAN

sama, *Game* saat ini telah dimainkan oleh banyak orang dari usia muda sampai tua. *Game* jenisnya semakin bervariasi sejalan dengan perkembangan teknologi perangkat keras dan perangkat lunak. Perkembangan *game* saat ini berkembang pesat, saat ini tidak hanya dimainkan di komputer, yang hanya bisa dimainkan dengan teman yang dikenal, tetapi saat ini sudah banyak *game* yang dimainkan di dunia maya yang sering disebut dengan *game on-line* yang dimainkan oleh banyak orang dalam satu permainan walau jarak mereka saling berjauhan dan tidak saling mengenal. Pada awalnya *game* identik dengan anak – anak. Para orang dewasa selalu berpikir *game* merupakan suatu kegiatan yang dilakukan oleh anak – anak yang dapat menyenangkan hati mereka.

Dengan kata lain, segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai *game*. *Side Scrolling Game* adalah salah satu *genre game* dengan sudut pandang kamera seolah – olah di samping karakter, dan mengikuti pergerakan pemain yang pada umumnya bergerak dari sisi kiri menuju sisi kanan layar untuk mencapai target atau lokasi yang di tentukan. Alasan menerapkan *side scrolling game* pada *game* yang akan di bangun adalah, karena *game* sejenis ini paling cocok untuk diterapkan pada umur 6 tahun ke atas, karena kontrol permainannya yang tidak terlalu rumit dalam memainkannya.

Berdasarkan latar belakang tersebut, maka diambil tema skripsi ini dengan judul “Membangun *Side Scrolling Game Grim Reaper Flying In The Darkness*” dengan menggunakan algoritma *collision detection* sehingga dapat mendeteksi benturan antara satu dengan objek lainnya.

2. RUANG LINGKUP PENELITIAN

Permasalahan difokuskan pada :

1. Membangun *game grim reaper flying in the darkness* dengan algoritma *collision detection* sehingga dengan algoritma tersebut benturan antara objek satu dan objek lainnya terlihat lebih nyata.
2. Permainan ini menerapkan pengacakan pada gerak musuh agar pergerakan musuh tidak monoton.
3. Membuat permainan dengan *genre side scrolling game*. *Genre* yang sangat cocok untuk dimainkan untuk usia 6 tahun ke atas karena kontrol *game*.
4. Membuat permainan yang dinamis dan tidak membosankan dengan tampilan yang menarik.

3. BAHAN DAN METODE

Adapun bahan dan metode yang gunakan dalam membangun *game* ini yaitu:

3.1 Algoritma Pengacakan Posisi Objek

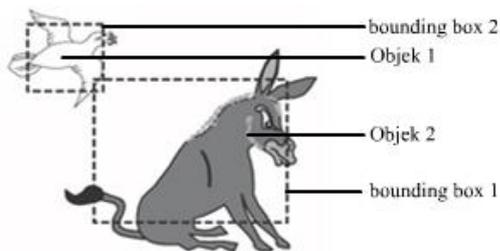
Menurut Schwarzl, (2012), *Collision detection* adalah proses pendeteksian tabrakan antara dua objek. Sebenarnya dalam *game* sendiri tabrakan tidak hanya terjadi antara dua objek, tetapi dapat terjadi juga antara

satu objek dengan banyak objek atau banyak objek dengan banyak objek. Dalam game membutuhkan *collision detection* yang akurat karena setelah menentukan terjadinya collision atau tidak kita harus menentukan apa yang terjadi pada objek yang ditabrak ataupun yang menabrak. *Collision detection* ini juga berguna untuk menentukan posisi dari satu objek dengan objek yang lain sehingga tidak ada objek yang saling menembus. Sehingga game yang akan dibuat memiliki kesamaan dengan realita yang ada. *Collision Detection* memiliki teknik yang berbeda – beda untuk diterapkan pada game 2D. Teknik *collision Detection* pada 2D ada 2 cara yaitu :

1. *Rectangles Collision detection*
2. *Circle rectangle collision detection*

Rectangles Collision Detection merupakan pendeteksi tabrakan menggunakan *bound* berbentuk persegi, dimana setiap *sprite* atau objek yang dibuat dalam game memiliki *bound* masing – masing . Cara ini cukup mudah karena hanya perlu membandingkan apakah ada *intersection* (irisan) pada kedua objek pada koordinat tertentu. Adapun cara lain dari hasil modifikasi teknik ini yaitu dengan *reduced size bounding box*. Dimana kotak batasnya diperkecil dengan mengurangi beberapa pixel dari ukuran *sprite* atau objek yang sebenarnya. Sehingga dengan cara ini pada saat melakukan *collision detection* akan lebih akurat karena ruang kosong antara *sprite* dan *bound* berkurang.

Kemudian cara lain dari modifikasi teknik ini adalah dengan cara menggunakan beberapa *bounding box* secara bersamaan untuk satu *sprite* atau objek. Contohnya untuk *sprite* berbentuk manusia, pertama *bounding box* digunakan untuk bagian kepala kemudian *bounding box* yang lainnya digunakan untuk badan dan kaki. Cara ini lebih akurat karena banyaknya *bounding box* yang digunakan dapat disesuaikan dengan bentuk gambar pada *sprite*.



Gambar 1. Reduced Size Bounding Box Collision Detection.

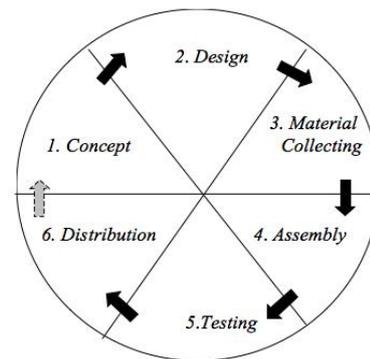
penerapan algoritma *reduced size bounding box Collision Detection* pada gambar 2.1 menunjukkan bahwa jika objek 1 yang telah disisipkan *bounding box 1* bertabrakan pada objek 2 yang telah disisipkan *bounding box 2* maka *Collision* bernilai *true* jika tidak maka *collision* bernilai *false*. Dibawah adalah contoh script dengan menggunakan algoritma *reduced size bounding box collision detection*.

```
If (objek1.boundingBox1.hitTest(objek2.boundingBox2)) {
    collision = true;
```

```
}else{
    collision = false;
}
```

3.2 Tahapan Pengembangan Multimedia

Menurut Binanto (2010), metodologi pengembangan multimedia terdiri dari enam tahap, yaitu *concept* (pengonsepan), *design* (pendesainan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Keenam tahap ini tidak dapat bertukar posisi. Meskipun begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan.



Gambar 2. Tahapan Pengembangan Multimedia

Tahapan Pengembangan Multimedia Meliputi :

1. Concept

Tahapan *concept* (pengonsepan) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi *audiens*). Tujuan dan pengguna akhir program berpengaruh pada nuansa multimedia sebagai pencerminan dari identitas organisasi yang menginginkan informasi sampai pada pengguna akhir. Karakteristik pengguna termasuk kemampuan pengguna juga perlu dipertimbangkan karena dapat memengaruhi pembuatan desain.

Selain itu, tahap ini juga akan menentukan jenis aplikasi (presentasi, interaktif, dan lain-lain) dan tujuan aplikasi (hiburan, pelatihan, pembelajaran dan lain-lain). Dasar aturan untuk perancangan juga ditentukan pada tahap ini, misalnya ukuran aplikasi, target, dan lain-lain. *Output* dari tahap ini biasanya berupa dokumen yang bersifat naratif untuk mengungkapkan tujuan proyek yang ingin dicapai.

2. Design

Design (perancangan) adalah tahap pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan, dan kebutuhan material/bahan untuk program. Spesifikasi dibuat serinci mungkin sehingga pada tahap berikutnya, yaitu *material collecting* dan *assembly*, pengambil keputusan baru tidak diperlukan lagi, cukup menggunakan keputusan yang sudah ditentukan pada tahap ini. Meskipun demikian, pada prakteknya, pekerjaan proyek pada tahap awal masih akan sering mengalami penambahan bahan atau pengurangan bagian aplikasi, atau perubahan-perubahan lain.

3. Material Collecting

Material Collecting adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan. Bahan-bahan tersebut, antara lain gambar *clip art*, foto, animasi, *video*, *audio*, dan lain-lain yang dapat diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangannya. Tahap ini dapat dikerjakan secara paralel dengan tahap *assembly*. Namun, pada beberapa kasus, tahap *material collecting* dan tahap *assembly* akan dikerjakan secara linear dan tidak paralel.

4. Assembly

Tahap *Assembly* adalah tahap pembuatan semua objek atau bahan multimedia. Pembuatan aplikasi didasarkan pada tahap *design*, bagan alir, dan /atau struktur navigasi.

5. Testing

Tahap *Testing* (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan melihatnya apakah ada kesalahan atau tidak. Tahap pertama pada tahap ini disebut tahap pengujian *alpha* (*alpha test*) yang pengujiannya dilakukan oleh pembuat atau lingkungan pembuatnya sendiri. Setelah lolos dari pengujian *alpha*, pengujian *beta* yang melibatkan penggunaan akhir akan dilakukan.

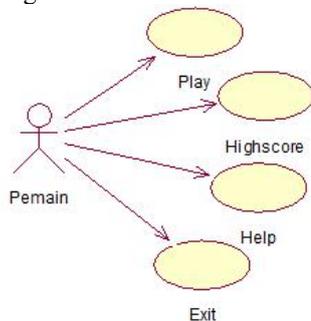
6. Distribution

Pada tahap ini, aplikasi akan disimpan dalam suatu media penyimpanan. Jika media penyimpanan tidak cukup untuk menampung aplikasinya, kompresi terhadap aplikasi tersebut akan dilakukan. Tahap ini juga dapat disebut tahap evaluasi untuk pengembangan produk yang sudah jadi supaya menjadi lebih baik. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap *concept* pada produk selanjutnya.

4. RANCANGAN SISTEM/APLIKASI

Analisis dan perancangan sistem *Grim Reaper Flying In The Darkness* adalah sebagai berikut :

1. Usecase Diagram

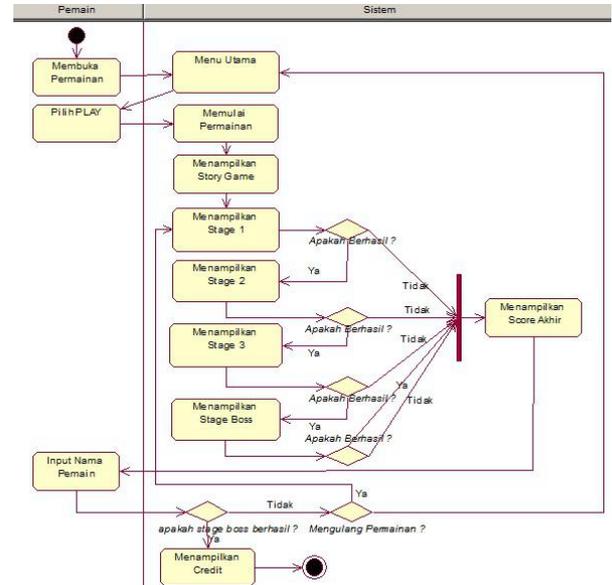


Gambar 3. Usecase Diagram Grim Reaper Flying In The Darkness

Pada gambar 3 dapat dilihat bahwa pemain dapat melakukan beberapa aksi sebelum memulai permainan. Dalam menu utama pemain dapat memilih "Play" untuk memulai permainan, "Highscore" Menampilkan Papan

Highscore (*Leader Board*) antara pemain lain. "Help" Menampilkan bantuan untuk pemain, dan "Exit" Keluar dari permainan.

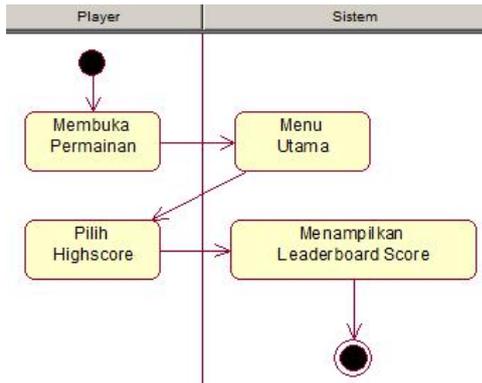
2. Activity Diagram



Gambar 4. Activity Diagram untuk "PLAY" pada Grim Reaper Flying In The Darkness

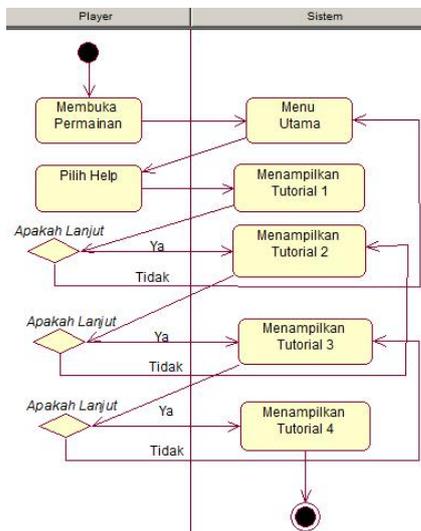
Pada gambar 4 dapat dilihat alur aktifitas yang terjadi disaat pemain memilih "PLAY" pada menu utama sampai menyelesaikan permainan. Sistem akan menampilkan menu utama dimana pemain memilih "PLAY" untuk memulai permainan, kemudian sistem akan menampilkan *Stage* Pertama lalu *Stage* pertama permainan dimulai.

Selanjutnya pemain memainkan *Stage* tersebut, Saat pemain gagal atau kalah pada *Stage* tersebut, Sistem akan menampilkan sebuah *Frame* Kalah atau *Game Over* yang didalamnya terdapat *Score* akhir permainan. Di dalam *Frame* Kalah ditampilkan dua pilihan yakni untuk mengulang atau kembali ke menu utama. Jika pemain memilih "Retry" maka sistem akan mengulang kembali *Stage* permainan yang sama. Dan jika pemain memilih "Back Menu" maka sistem akan menampilkan kembali menu awal permainan. Jika pemain berhasil memenangkan *Stage* permainan tersebut, Sistem akan menampilkan sebuah *Stage* yang akan mempertemukan Pemain dengan Boss Permainan. Dan jika pemain berhasil mengalahkan Boss maka Sistem akan menampilkan Sebuah *Scene* "Credit" yang menyatakan bahwa permainan telah selesai.



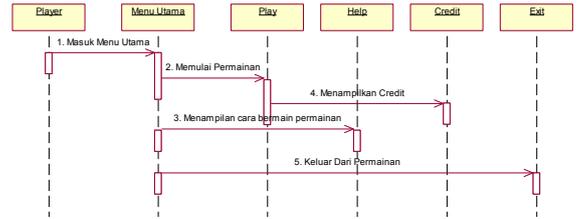
Gambar 5. Activity Diagram untuk “Highscore”

Pada gambar 5 dijelaskan alur aktifitas yang terjadi di saat pemain memilih “HIGHSCORE” pada menu utama dimana berisi *Leaderboard* atau score – score pemain lain, dimana terdapat urutan teratas merupakan score tertinggi (*Highscore*).



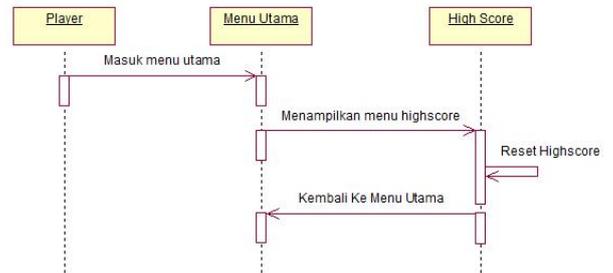
Gambar 6. Activity Diagram untuk “Help”

Pada gambar 6 dijelaskan alur aktifitas yang terjadi di saat pemain memilih “HELP” pada menu utama dimana berisi *tutorial* cara bermain permainan *Grim Reaper Flying In The Darkness*. Dimana Jika pemain memilih “Help 1” maka sistem akan menampilkan “Help 1” jika pemain melanjutkan ke “Help 2” Maka Sistem akan menampilkan “Help 2” Jika tidak maka sistem akan menampilkan menu utama. Jika pemain melanjutkan “Help 2” maka sistem akan menampilkan ke “Help 3” jika tidak maka Sistem akan menampilkan “Help 1” Jika pemain melanjutkan “Help 3” maka sistem akan menampilkan “Help 4” Jika tidak sistem akan menampilkan “Help 2”.



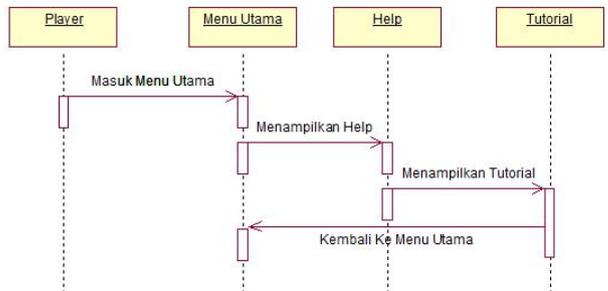
Gambar 7. Sequence Diagram *Grim Reaper Flying In The Darkness*

Gambar 7 menunjukkan proses yang terjadi antara pemain dengan sistem, dimana sistem mengeksekusi setiap proses permainan agar permainan dapat berjalan sesuai dengan semestinya.



Gambar 8. Sequence Diagram *Highscore* pada *Grim Reaper Flying In The Darkness*

Gambar 8 menunjukkan proses yang terjadi antara pemain dengan sistem saat player memilih menu *highscore*.



Gambar 9. Sequence Diagram *Highscore* pada *Grim Reaper Flying In The Darkness*

5. IMPLEMENTASI

Hasil implementasi berdasarkan analisis dan perancangan adalah sebagai berikut :

1. Tampilan *Scene* Menu Utama



Gambar 10. *Scene* Menu Utama

Pada gambar 10 Scene menu utama merupakan tampilan utama dari permainan *Grim Reaper Flying In The Darkness*. Disaat pemain baru memasuki sistem. Terdapat nama atau judul dari permainan “*Grim Reaper Flying In The Darkness*”. Setiap tombol pada Scene menu utama memiliki fungsi masing – masing yang mengabungkan satu scene ke scene lainnya. Tampilan Scene Menu Utama.

2. Scene Menu Play Stage 1



Gambar 11. Scene Menu Play Stage 1

Pada Gambar 11 Merupakan scene Stage 1 dimana dalam permainan grim reaper mendapatkan tugas untuk mendapatkan score sebanyak 130. Jika tidak menyelesaikan misi pemain tidak dapat melanjutkan ke stage

3. Scene Highscore



Gambar 12. Scene Highscore

Pada Gambar 12 merupakan Scene menu Highscore dalam game *Grim Reaper Flying In The Darkness*. Dimana terdapat kumpulan 5 player terbaik dalam memainkan game ini.

4. Scene Help Jump Tutorial



Gambar 13. Scene Help Jump Tutorial

Pada gambar 13 dijelaskan cara meloncat dalam game *Grim Reaper Flying In The Darkness*. Terdapat 2 loncatan dalam game ini. Yaitu loncatan sekali, dan loncatan dua kali. Dimana loncatan dua kali berguna juga untuk menyerang musuh tanpa menekan tombol pukul.

6. KESIMPULAN

Berdasarkan uraian dari masing- masing bab dan hasil pembahasan maka dapat disimpulkan bahwa pembuatan *Edugame Boni Kids – Borneo Animal Kids* sebagai berikut :

1. Telah dihasilkan sebuah permainan bergenre *Side Scrolling Game* ,*Grim Reaper Flying In The Darkness* yang telah terkomputerisasi dalam *Game Playnya*
2. Permainan *Grim Reaper Flying In The Darkness* ini menggunakan algoritma *Collision Detection* untuk pendeteksian tubrukan antara objek satu dengan objek lainnya
3. Berdasarkan hasil uji coba untuk proses tubrukan antara objek satu dengan objek lainnya dapat disimpulkan bahwa hasil perhitungan sistem sesuai dengan algoritma *Game Rulanya*
4. Berdasarkan hasil kesimpulan pengujian beta testing dapat disimpulkan bahwa yang mengatakan judul game sangat menarik ialah 70%, cerita dalam game menarik ialah 60%, desain game sangat menarik ialah 50%, sistem permainan yang digunakan menarik 60%,desain karakter objek menarik sangat menarik 50%, sound game menarik 70% dan game sangat menantang ialah 70%.
5. Berdasarkan hasil kesimpulan Pengujian Blackbox maka dapat disimpulkan hasil pengimputan dalam game *Grim Reaper Flying In The Darkness* sesuai.

7. SARAN

Berdasarkan hasil dari penelitian ini akan beberapa saran, yaitu sebagai berikut :

1. Perlu adanya perbaikan maupun penambahan fitur seperti tampilan berbasis 3D sehingga game tersebut dapat dilihat lebih menarik.
2. Permainan ini hanya berbasis game PC, diharapkan untuk penelitian selanjutnya ada mahasiswa yang mampu mengembangkan game menjadi berbasis *android*.
3. Diharapkan agar permainan ini dapat dijadikan sebagai acuan atau referensi bagi mahasiswa informatika dalam membuat tugas akhir.

8. DAFTAR PUSTAKA

- Anggra, 2008, *Memahami Teknik Dasar Pembuatan Game Berbasis Flash*, Yogyakarta : Gava Media.
- Binanto, Iwan, 2010, *Multimedia Digital-Dasar Teori dan Pengembangannya*, Yogyakarta: Andi
- Hariani, Dwi, 2011, *Pembuatan Game Side Scrolling Catch the apple menggunakan Adobe Flash CS3*. Yogyakarta : Skripsi S1 Sistem Informasi STMIK Amikom.
- Hendratman, Hendi, 2008, *The Magic of Adobe Photoshop*, Bandung: Informatika

- Jasson, 2009, *Role Playing Game (RPG) Maker*, Yogyakarta : Andi.
- Jounghyoun Kim, Gerrard. (2005). *Designing Virtual Reality System the structured approach*, USA : Springer.
- Nugraha, 2011, *Penggunaan Struktur Data Quad-Tree dalam Algoritma Collision Detection pada Vertical Shooter Game*, Makalah IF3051 Strategi Algoritma, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Bandung.
- Nugroho, Adi, 2005, *Fokus Bangun Dasar Perancangan Sistem Dengan UML*, Yogyakarta : Gramedia Pustaka Utama.
- Paiman, 2012, *Pembuatan Game 2D Jump And Run Side Scrolling Game "Ninja Yusuke"* Menggunakan GTGE. Yogyakarta : Skripsi S1 Teknik Informatika STMIK Amikom.
- Pressman, R, 2005, *Software Engineering : A Practitioner's Approach*. McGraw-Hill.
- Rizki, Cahya, 2013, *Perancangan Aplikasi Belajar Hijaiyah Pada Android Menggunakan Metode Rectangles Collision Detection*, Sumatera Utara : Skripsi S1 Teknik Informatika Universitas Sumatera Utara.
- Schwarzl, T, 2012. *2D Game Collision Detection*, USA : Createspace Independent Publishing Platform.