

PERANCANGAN APLIKASI GAME FIRST PERSON SHOOTING

Yulia Rachman, Patah Herwanto
greenwind.hayate@rocketmail.com
pherwanto@yahoo.com

ABSTRAK

Tak bisa dipungkiri bahwa aplikasi permainan atau *game* kini telah menjadi bagian dari masyarakat. Baik *game* yang paling sederhana sekalipun sampai *game* dengan rancangan yang paling rumit, ia menjadi sesuatu yang digemari oleh semua kalangan. Inilah yang menjadikan dunia *game* memiliki peluang bisnis sehingga para pengembang *game* terus bermunculan dan menciptakan *game* yang semakin menarik. Permainan penembak pertama atau lebih dikenal dengan *First Person Shooting* adalah salah satu dari sekian banyak jenis permainan yang memiliki banyak peminat. Namun, tentu saja karena melibatkan senjata dalam permainan ini, tak jarang pula unsur kekerasan disisipkan kedalamnya. Inilah sebabnya *game* tidak selalu dipandang sebagai hal yang positif. Untuk alasan itulah, kita merancang sebuah aplikasi permainan *first person shooting* yang sederhana yang lebih menekankan pada fokus pemain, daya tanggap, dan pada beberapa level memerlukan daya ingat yang baik dari pemain.

Dalam pembangunan *game* ini, menggunakan Software Adobe Flash CS6 dan memanfaatkan *actionscript* untuk mengatur semua elemen yang diperlukan didalamnya agar *game* tersebut dapat berjalan sesuai dengan semestinya. Dalam pengembangan *game* ini digunakan metode prototipe dimana ia merupakan metode yang telah umum digunakan oleh para pengembang aplikasi permainan.

Kata kunci : Aplikasi permainan, penembak pertama, Adobe Flash CS6, Actionscript, Prototipe.

1. PENDAHULUAN

1.1 Latar Belakang

Pengaplikasian ilmu komputer telah menghadirkan teknologi ke dalam kehidupan masyarakat, yang pada awalnya diciptakan untuk membantu memudahkan manusia dalam menyelesaikan suatu pekerjaan. Seiring dengan berjalannya waktu, kebutuhan yang terus meningkat telah memacu perkembangan teknologi. Hal inilah yang telah memperluas fungsi teknologi komputer. Ia tidak hanya digunakan untuk mengolah data, informasi, sebagai alat untuk membantu membuat keputusan, sebagai media komunikasi dan lain sebagainya, selain itu penerapan ilmu komputer juga diaplikasikan dalam dunia hiburan, misalnya aplikasi untuk musik, animasi, video, dan juga *game*.

Perkembangan aplikasi permainan yang sangat cepat dan hampir tidak ada batasan dalam perancangannya ini menyebabkan para *developer* terus melakukan peremajaan sehingga aplikasi ini mampu menyerap lebih banyak perhatian publik. Dunia *game*

selalu mempunyai tempat di masyarakat. Jenisnya sangat beragam, dan alur permainan apapun tidak pernah membosankan penggunaannya. Tidak terhitung berapa kali inovasi yang telah dilakukan serta berapa banyak hasil karya yang telah dipublikasikan dan sukses di pasaran, karena itulah industri *game* atau *game developer* terus bermunculan dan berkembang dengan sangat pesat dari waktu ke waktu. Peminatnya berasal dari berbagai kalangan dan usia sehingga membuat industri *game* bisa saja menjadi peluang bisnis yang menjanjikan.

1.2 Identifikasi Masalah

Sebuah *game* tidak dibuat begitu saja tanpa tujuan, dan tidak semudah yang dibayangkan. Perancangannya harus berdasarkan pada perhitungan yang matang agar hasilnya memuaskan. Masalah yang timbul dalam pembuatan *game* adalah :

1. Bagaimana menerapkan ilmu komputer untuk berbagai tujuan, termasuk untuk tujuan *entertainment* dan komersial
2. Tidak terbatasnya pengembangan aplikasi permainan, secara tidak langsung memperbolehkan unsur apapun untuk dimasukkan ke dalam *game* demi menarik minat yang besar . Tetapi tak jarang para *game developer* memasukkan unsur negatif dalam aplikasi permainannya. Inilah yang menjadi akar permasalahan mengapa *game* kerap kali dipandang tidak memberikan dampak yang positif.
3. Bagaimana membuat sebuah *game* yang dapat diterima oleh masyarakat

2. METODOLOGI

Dalam merancang sebuah aplikasi permainan untuk *personal computer* ini, menggunakan metode *prototyping*. *Prototyping* adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau komponen-komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan. (Howard, 1997) *Prototype* diklasifikasikan berdasarkan penggunaan dan levelnya. Berdasarkan pada penggunaannya, hasil akhir *prototype* diklasifikasikan menjadi dua, yaitu:

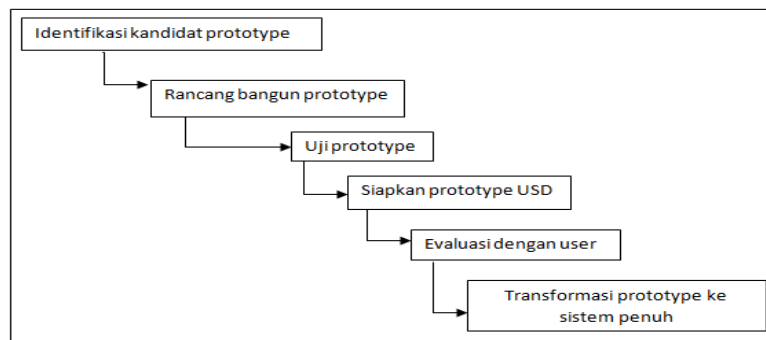
1. *Reusable Prototype*, yaitu *prototype* yang akan ditransformasikan menjadi produk final.

2. *Throwaway prototype*, adalah *prototype* yang akan dibuang atau tidak akan digunakan lagi setelah selesai menjalankan tujuan dibuatnya *prototype* tersebut.

Sedangkan berdasarkan pada levelnya, *prototype* diklasifikasikan menjadi :

1. *Input/output Prototype*, yaitu *prototype* yang dibuat hanya terbatas pada *user interface* atau antar muka pengguna.
2. *Processing prototype*, yakni *prototype* yang meliputi perawatan file dasar dan proses-proses.
3. *System prototype*, adalah *prototype* yang berupa model lengkap dari perangkat lunak.

Dalam *prototyping*, tahapan yang dikerjakan adalah bagian yang mudah terlebih dahulu, dan kemudian sesegera mungkin sampaikan hasil kepada pengguna. Dengan begitu, dapat mengetahui apa saja kebutuhan pengguna, apakah harus ada pengembangan sistem, atau menghapus bagian-bagian yang tidak dikehendaki oleh pengguna. Tahapan-tahapan *prototyping* bisa dilihat dari gambar berikut :



Gambar 1: Prototype model (Harris,2003)

Tahapan-tahapan dalam model *prototype* tersebut dapat dijelaskan sebagai berikut:

1. Identifikasi kandidat *prototype*.

Yang dimaksud kandidat dalam *prototype* yaitu meliputi *user interface* (termasuk di dalamnya menu, dialog, input, dan output), file, atau fungsi-fungsi pemrosesan. Dalam proses ini diambil beberapa sampel atau kandidat *prototype*, kemudian dipilih *prototype* mana yang paling mendekati ide atau rancangan dari aplikasi yang ingin dibuat.

2. Rancang bangun *prototype*
Setelah pemilihan kandidat *prototype*, maka langkah selanjutnya adalah mulai merancang dan membangun *prototype* tersebut dengan bantuan *software* yang mendukung dalam pembuatan *game*.
3. Uji *prototype*
Uji *prototype* untuk memastikan *prototype* dapat dengan mudah dijalankan untuk tujuan demonstrasi.
4. Siapkan *prototype* USD (*User's System Diagram*)
Siapkan *Prototype User's Diagram System* untuk mengidentifikasi bagian-bagian dari perangkat lunak yang di-*prototype*-kan.
5. Evaluasi dengan user
Langkah selanjutnya adalah mengevaluasi dengan user untuk mengidentifikasi dan melakukan perubahan jika diperlukan.
6. Transformasi *prototype* ke sistem penuh
Langkah akhir adalah mentransformasikan *prototype* menjadi perangkat lunak yang beroperasi penuh dengan melakukan penghilangan kode-kode yang tidak dibutuhkan, penambahan bagian-bagian yang memang dibutuhkan dan perbaikan, serta melakukan pengujian perangkat lunak secara berulang.

3. PEMBAHASAN

3.1 Identifikasi Kandidat *Prototype*

Ide konsep dari permainan yang akan dibuat adalah, pemain harus menembak target dengan jumlah tertentu dalam waktu yang terbatas. Jika pemain berhasil memenuhi jumlah target tersebut, maka pemain akan menang. Tapi jika pemain tidak bisa menembak target dalam jumlah yang cukup sampai batas waktu (*deadline*) habis, maka pemain akan kalah dan harus mengulang lagi permainannya dari awal.

Setelah menentukan ide *game* apa yang akan dibuat, maka tahap selanjutnya adalah mengidentifikasi kandidat *prototype*, dimana pada tahapan ini akan dipilih aplikasi *game* yang sebenarnya sudah ada, yang paling mendekati ide dari *game* yang akan dibuat. *Game* tersebut kemudian diambil untuk dijadikan *prototype* awal, lalu diidentifikasi apa saja kelebihan, kekurangan, dan hal-hal yang diinginkan *user* untuk ditambahkan atau dikurangi demi perkembangan *game* tersebut.

3.2 Rancang Bangun Prototype

Tahapan umum dalam perancangan *game* yang termasuk ke dalam tahapan rancang dan bangun prototype adalah *design treatment*, *assets developmet phase*, dan *programming phase*.

3.2.1 Design Treatment

Design treatment adalah sebuah istilah dalam *game development* yang bapat berarti sebuah desain yang mengandung hasil analisa mendalam terhadap aplikasi *game* yang dirancang, termasuk didalamnya adalah analisis sistem, penggambaran semua misi yang harus dilakukan pemain untuk merampungkan keseluruhan permainan, desain *input-output* dan juga penentuan tampilan visual yang akan digunakan dalam aplikasi *game*.

3.3 Analisis Sistem

Analisis sistem mencakup analisa tentang keseluruhan game. Mulai dari analisis game, analisis pengguna, analisis kebutuhan perangkat lunak, dan analisis kebutuhan perangkat keras.

3.3.1 Analisis Game

Berikut adalah analisis tentang game yang akan dibuat.

1. Kategori : Game untuk *personal computer*
2. *Genre* : *First Person Shooting* (FPS)
3. Setting : Mengambil setting di tempat pelatihan menembak dengan nuansa dan ruang yang berbeda di setiap levelnya.
4. Level : Terdapat 7 misi yang harus diselesaikan dengan target dan tingkat kesulitan yang berbeda-beda.
5. Visualisasi : Visualisasi grafis yang akan digunakan ialah grafis 2D (dua dimensi)
6. Audio : *Game* ini menggunakan efek suara dengan format Wav

3.3.2 Analisis Pengguna

Target audience adalah masyarakat yang akan menggunakan aplikasi *game* yang akan dibuat. Dalam *Game First Person Shooting* ini, maka *target audience* nya ditujukan untuk masyarakat usia 10 tahun ke atas. Kemudian tidak ada batasan bagi tingkat pendidikan. Siapapun yang bisa bermain *game* dan mengerti aturan mainnya maka ia boleh memainkan aplikasi *game* ini. Status sosial pengguna dalam masyarakat adalah kalangan menengah ke atas, yang memiliki komputer untuk bisa memainkan permainan ini. Selain itu, *game* ini dapat dimainkan oleh siapa saja yang menyukai jenis permainan seperti ini, baik laki-laki maupun perempuan. Jadi tidak ada batasan *gender* dalam permainannya. *Game* ini juga ditujukan bagi mereka yang membutuhkan suatu permainan yang dapat menghibur dan menghilangkan kejenuhan sehingga adanya kepuasan batin.

3.3.3 Analisis Kebutuhan Perangkat Lunak

Game First Person Shooting ini memiliki kebutuhan perangkat lunak sebagai berikut :

1. Adobe Flash Professional CS6, yang merupakan *software* yang dipakai untuk menghasilkan aplikasi berbasis Action Script.
2. Flash Player, adalah sebuah aplikasi yang diperlukan untuk menjalankan file bertipe SWF.

3.3.4 Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang dibutuhkan agar Game First Person Shooting ini berjalan dengan baik adalah:

1. Prosesor dengan kecepatan 1.8 GHz
2. Memori 256 Mb
3. Harddisk 20 GB
4. VGA card 256 Mb
5. Monitor
6. Mouse

3.4 Desain Level

Misi dalam *game* harus diuraikan lebih jelas dan terperinci agar konsep dari *game* dapat betul-betul dipahami sehingga dapat diketahui apa saja aset yang perlu dibuat untuk mendukung *game* tersebut. Kemudian harus ditentukan pula :

- a. *Roleplay* atau *Rules* , yang merupakan aturan main yang harus diikuti oleh pemain.
- b. *Reward*, yaitu penghargaan atau sesuatu yang didapatkan oleh pemain jika berhasil mencapai tujuan dari permainan.
- c. *Punishment*, adalah hukuman atau konsekuensi yang harus diterima oleh pemain ketika dinyatakan gagal atau jika bermain tidak sesuai dengan aturan.

Maka, *rules* dari permainan ini adalah , pemain harus menembak target sampai memenuhi bar. Peluru yang ada dalam pistol adalah 6 butir, jika habis maka dapat diisi ulang (*reload*) dengan bebas, tidak ada batasan berapa kali harus mengisi ulang. Waktu yang diperlukan ialah kurang dari dua puluh detik, atau sampai bar waktu habis. Namun pada misi tertentu, ada juga yang tidak memiliki batas waktu.

Reward dalam *game* ini adalah penambahan poin untuk setiap target yang tertembak dengan besar poin yang bervariasi. Pemain juga berhak untuk maju ke level berikutnya jika berhasil memenuhi bar target.

Punishment jika pemain tidak berhasil adalah tampilan *game* akan masuk ke dalam frame *time-out* (kehabisan waktu) dan harus mengulang terus permainannya sampai berhasil. Dalam misi tertentu, jika pemain menembak target yang salah, maka akan ada dua kemungkinan sanksi yaitu pengurangan poin, atau permainan langsung berhenti dan pemain diminta untuk mengulang dari level sebelumnya.

3.5 Desain Input Output

Dalam sebuah *game*, *input* (masukan) dan *output* (keluran), merupakan sebuah interaksi antara pemain dengan aplikasi permainan. Interaksi ini akan berjalan lancar jika responnya tepat, tidak terlalu lambat dan tidak terlalu cepat.

Dalam *game first person shooting* ini, input yang didapat dari pemain berupa penggunaan *mouse*. *Input* ini akan direspon dengan *output* berupa audio, visual dan animasi.

1. *Input* dengan mengarahkan mouse dan klik kiri pada ikon tombol, maka *output* yang dihasilkan ialah perpindahan frame satu kepada frame yang lain yang akan ditampilkan pada layar. Perubahan tampilan merupakan *output* dengan hasil visualisasi.
2. Ketika misi dimulai, pemain dapat menggerakkan *crosshair*, tangan dan senjata, merupakan respon dari *mouse* yang digerakkan sesuai kehendak pemain.
3. Ketika menembak, *input*-nya adalah klik kiri pada *mouse*, dan *output*-nya adalah respon berupa :
 - a. suara tembakan (audio),
 - b. pergerakan tangan dan senjata (animasi),
 - c. berkurangnya jumlah peluru pada indikator.
4. Saat pemain berhasil menembak target dengan input klik kiri pada target, maka *output*-nya adalah :
 - a. suara tembakan (audio),
 - b. pergerakan tangan dan senjata (animasi),
 - c. berkurangnya jumlah peluru pada indikator.
 - d. Tampilan animasi pada target, yang pada awalnya berbentuk utuh, menjadi pecah dan kemudian hilang,
 - e. Penambahan nilai yang akan ditampilkan pada papan skor,
 - f. Penambahan isi dari bar target.
5. Ketika pemain menembak target yang sebaiknya dihindari, dengan input klik kiri pada *mouse*, maka *output*-nya adalah :
 - a. Suara tembakan
 - b. Animasi senjata dan tangan,
 - c. Berkurangnya jumlah peluru pada indikator,
 - d. Animasi target yang hancur,
 - e. Pengurangan nilai yang ditampilkan pada layar.
6. Jika pemain mengarahkan *mouse* pada objek yang dilarang untuk ditembak, dan meng-klik kiri pada *mouse*, maka *output*-nya ialah :
 - a. Suara tembakan
 - b. Animasi senjata dan tangan,
 - c. Berkurangnya jumlah peluru pada indikator,

- d. Animasi objek,
- e. Pergantian frame.

3.6 Desain Sistem

Perancangan sistem ini dimodelkan dengan *Unified Modeling Language (UML)* yang terdiri dari *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram*. Penggambaran model UML ini dapat membantu mempermudah proses pemrograman. Selain itu, digambarkan pula *storyboard* untuk *game* yang akan dibuat.

3.6.1 Use Case Diagram

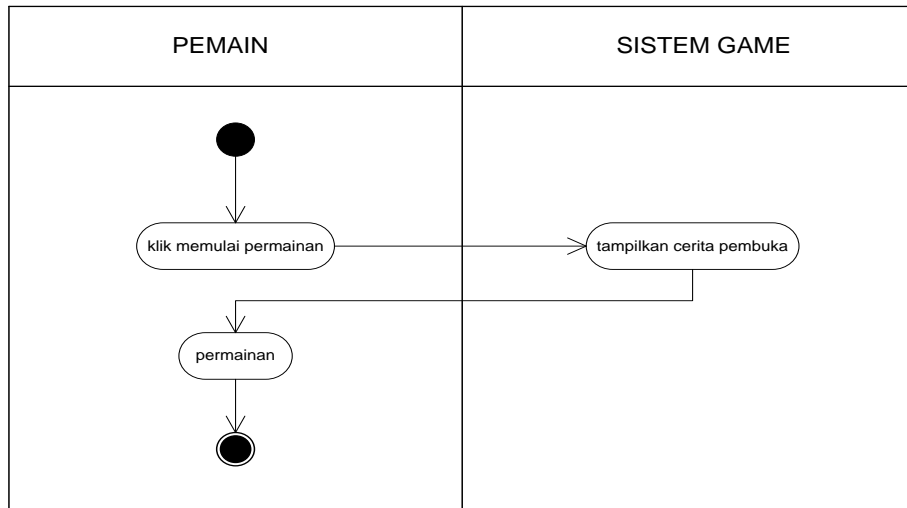
Use Case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan oleh aktor. *Use Case Diagram* menggambarkan pola perilaku sistem dan urutan kegiatan yang dilakukan oleh aktor.



Gambar 2 : Use Case Diagram

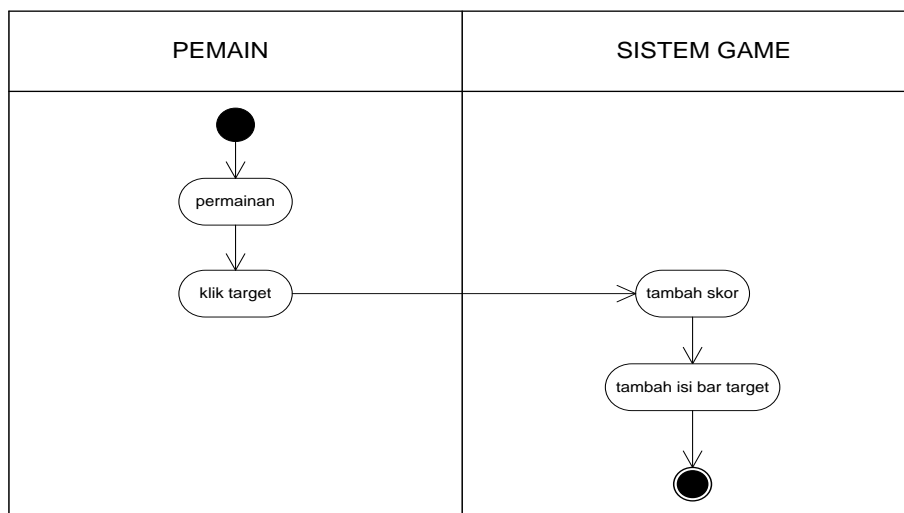
3.6.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, yaitu menggambarkan bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi dan bagaimana alir tersebut berakhir. Activity diagram dibuat untuk membantu memahami proses secara keseluruhan.



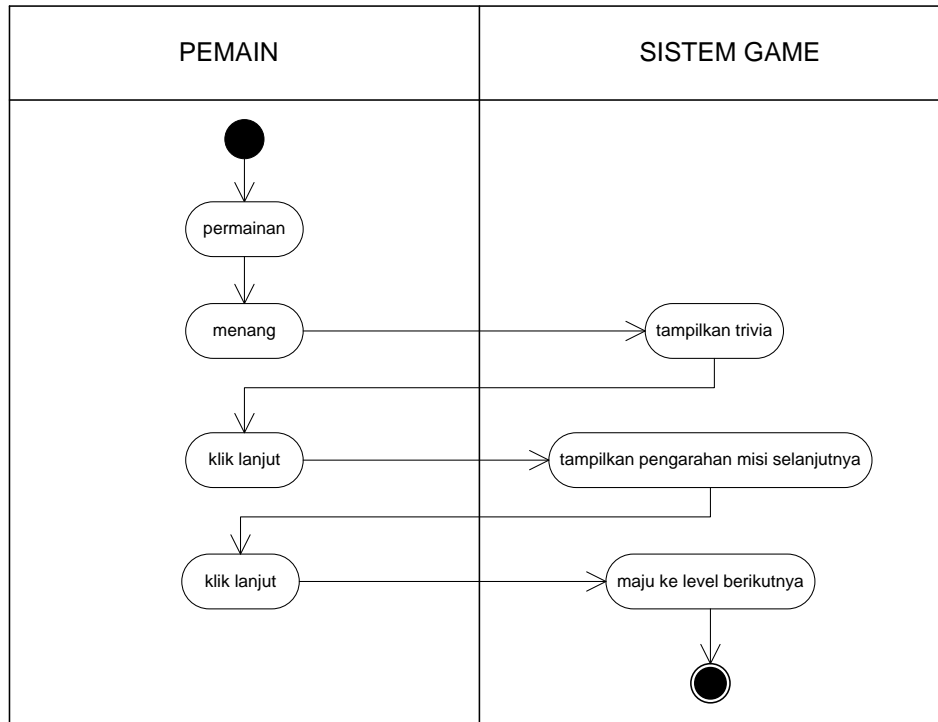
Gambar 3 : Activity Diagram memulai permainan

Activity Diagram memulai permainan menunjukkan bagaimana pemain memulai permainan dan akan direspon oleh sistem dengan menampilkan cerita pembuka, setelah itu menampilkan permainan kepada pemain.



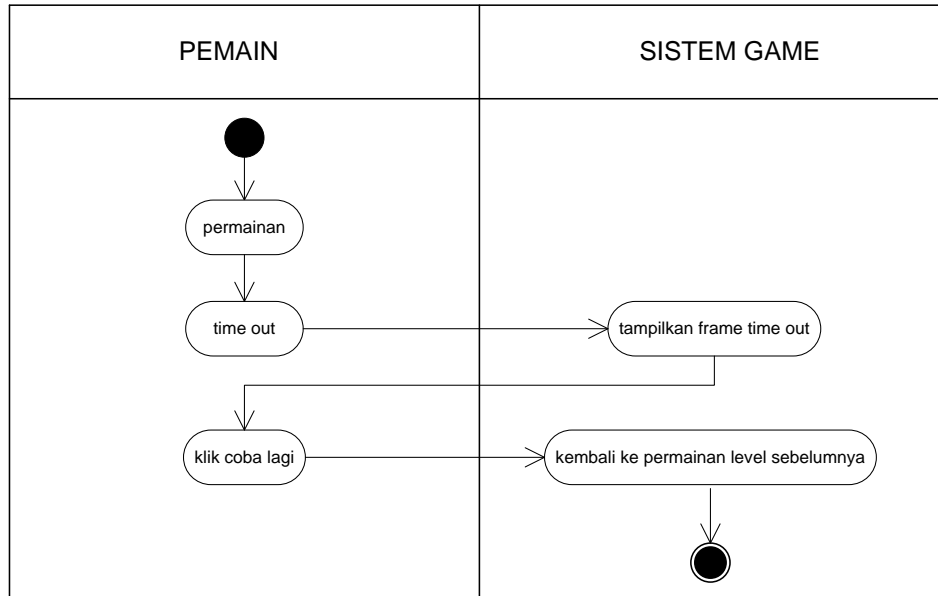
Gambar 4 : Activity Diagram tembak target

Activity diagram tembak target menguraikan tentang proses ketika permainan berlangsung. Dalam permainan, pemain dapat menembak target dengan klik pada target, sistem akan merespon dengan penambahan skor dan mengisi bar target apabila pemain berhasil menembak target.



Gambar 5 : Activity diagram menang

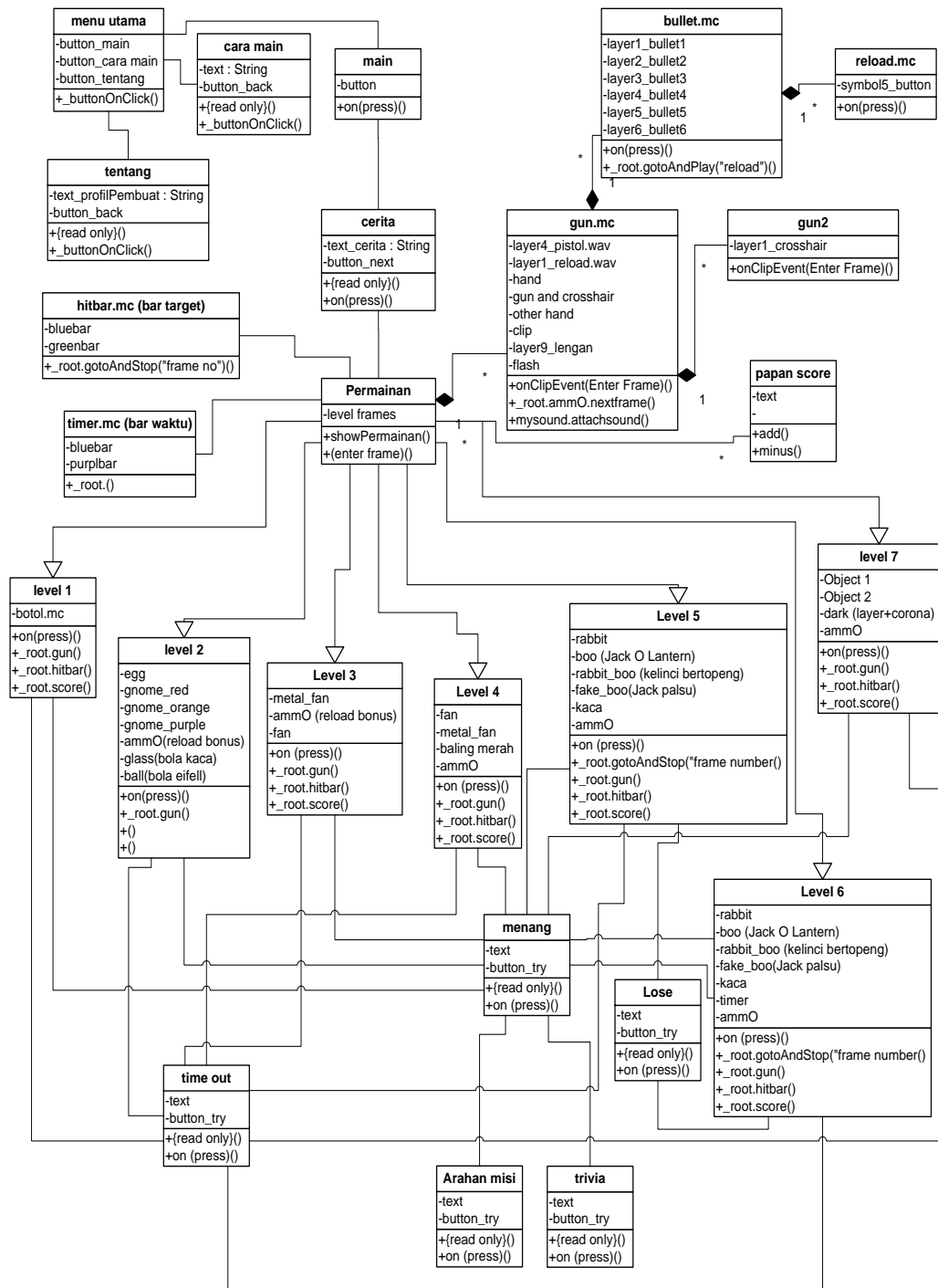
Activity Diagram diatas menguraikan ketika pemain berhasil memenangkan permainan, sistem akan menampilkan trivia, kemudian penjelasan mengenai misi pada level berikutnya. Baru kemudian akan ditampilkan permainan level berikutnya.



Gambar 6 : *Activity diagram time out*

Activity diagram time out, menjelaskan ketika pemain tidak bisa menyelesaikan permainan karena kehabisan waktu. Ketika waktu habis, sistem akan langsung memindahkan frame permainan pada frame time out. Tetapi pemain dapat mengulang kembali permainannya dengan klik pada tombol.

3.6.3 Class Diagram

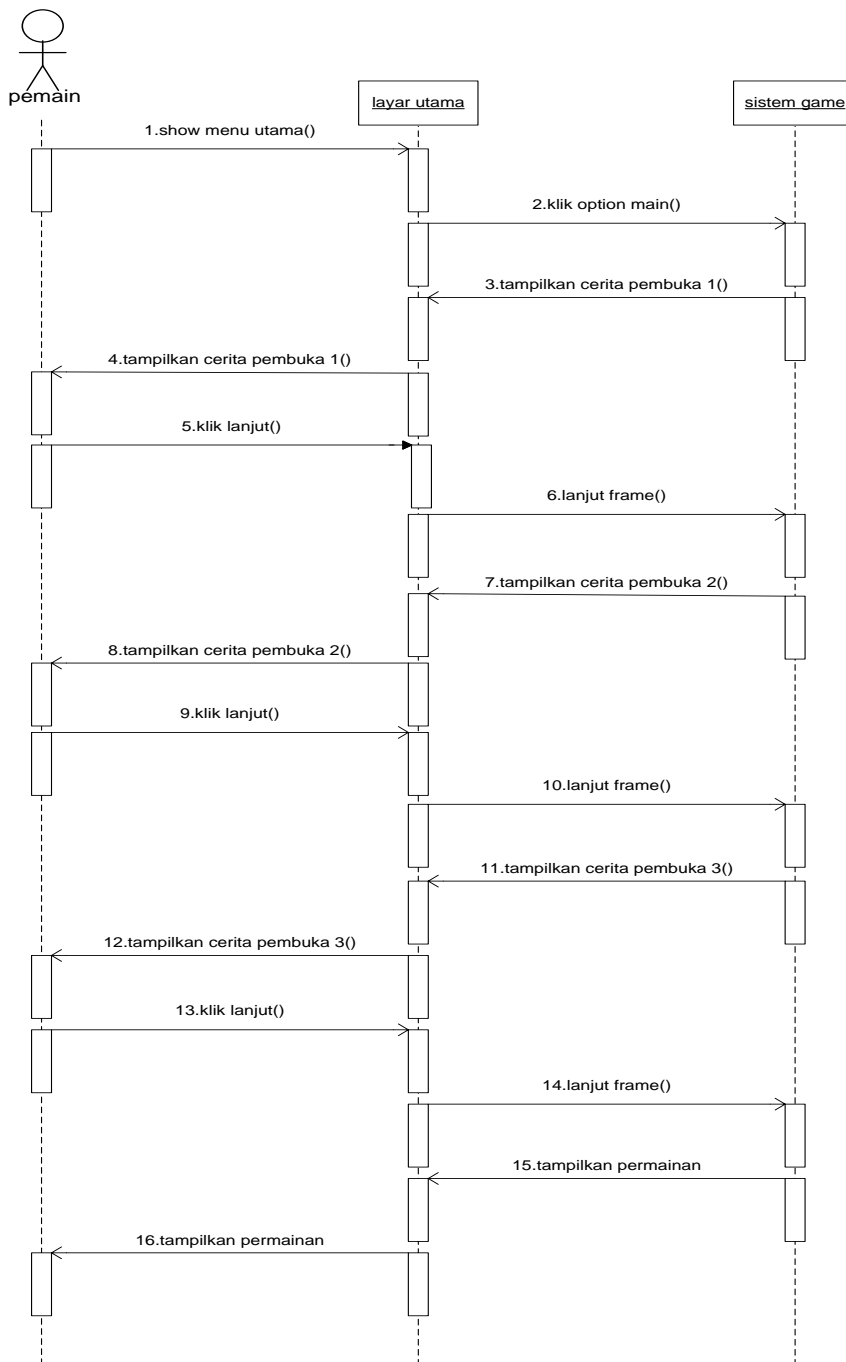


Gambar 7 : Class Diagram

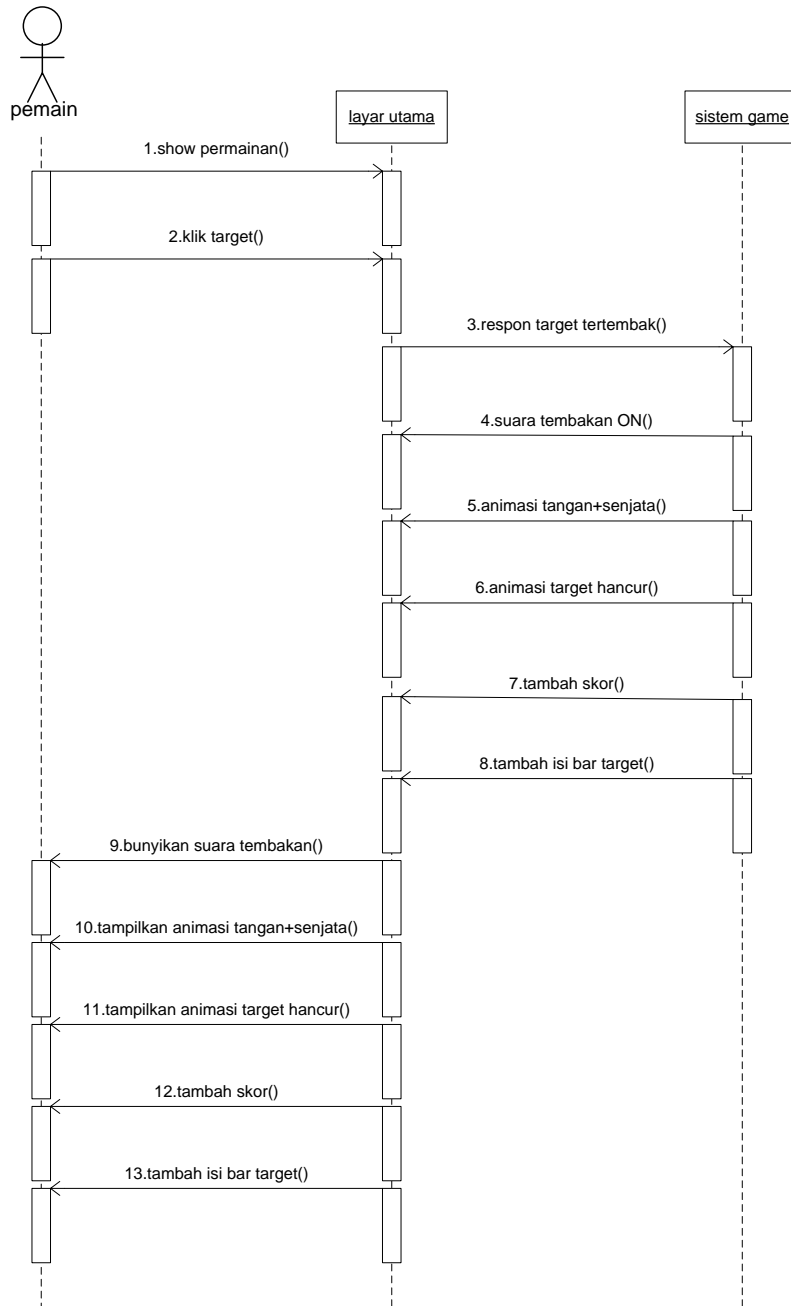
3.6.4 Sequence Diagram

Sequence diagram adalah suatu diagram yang menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Ia digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang

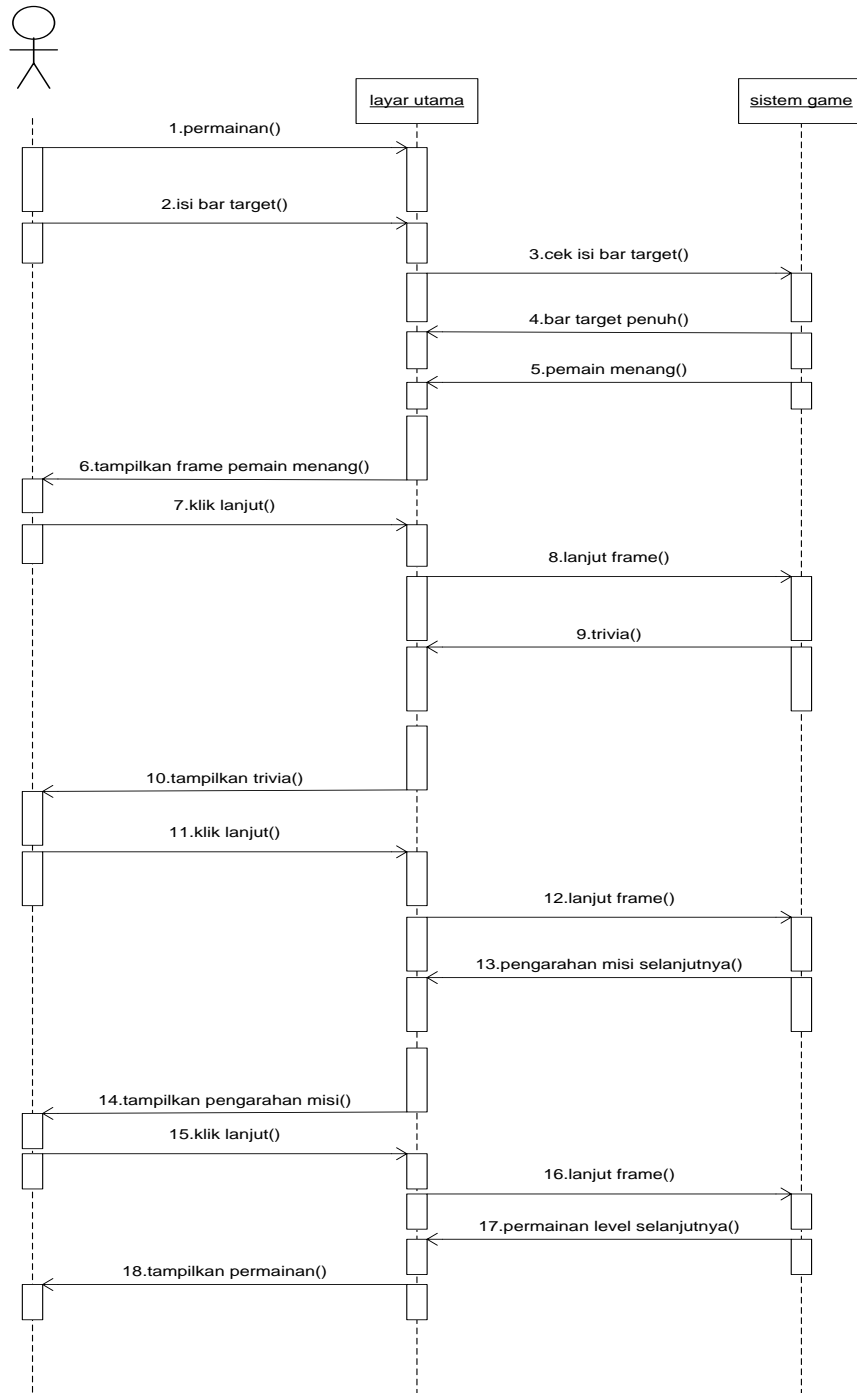
dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan *output* tertentu.



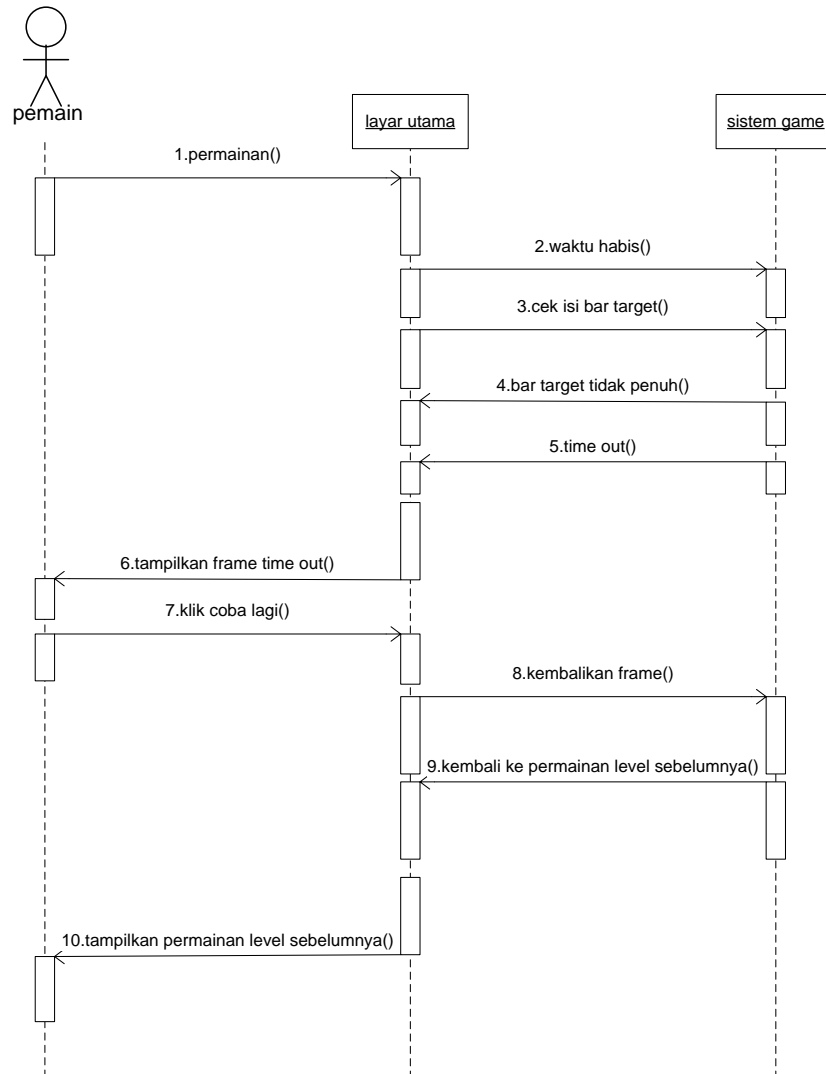
Gambar 8 : *Sequence diagram* memulai permainan



Gambar 9 : Sequence diagram tembak target



Gambar 10 : Sequence diagram menang



Gambar 11 : *Sequence Diagram time out*

3.6.5 Storyboard

Pembuatan *storyboard* penting dalam perancangan aplikasi multimedia, terutama sebuah *game*. *Storyboard* dalam *game* merupakan sebuah naskah yang disampaikan melalui sketsa gambar. Sketsa gambar ini merepresentasikan ide dari seorang *game developer* dan berfungsi untuk mempermudah orang lain untuk memahami ide-ide tersebut.

3.6.6 Asset Development Phase

Pada tahapan ini, aset-aset utama yang membangun sebuah *game* akan dibuat, meliputi aset visual, animasi dan audio.

Aset visual meliputi grafis karakter utama yang akan diperankan oleh pemain, tipografi atau penggunaan huruf, dan grafis *interface* yang digunakan sebagai antarmuka dalam *game*.

Dalam sebuah *game*, animasi menjadi elemen yang penting sebab tanpa animasi, *game* akan tampak sangat kaku dan tidak akan menarik bagi penggunanya. Dalam *game*, animasi merupakan respon dari input yang dimasukkan oleh pemain. Animasi diciptakan dengan memainkan beberapa frame-frame gambar yang dibuat secara manual. Perpindahan yang cepat antara frame kepada frame yang lainnya akan membuat sebuah objek tampak bergerak secara nyata. Perpindahan ini dinyatakan dalam fps (*frame per second*), maka jika animasi pada *game* ini adalah 20 fps, artinya dalam satu detik ada 20 frame yang dimainkan

Komponen audio adalah aset yang penting yang digunakan untuk mempertegas permainan dan mendukung efek grafis yang ada di dalam *game*.

3.6.7 Programming Phase

Pada tahapan ini mulailah dibuat kode agar *game* dapat berjalan sesuai dengan yang diharapkan. Ini adalah tahapan yang paling penting dan dibutuhkan ketelitian dalam pengerjannya, sebab kesalahan dalam penulisan kode akan menyebabkan *game* tidak bisa dikompilasi.

Adobe Flash Professional CS6 merupakan sebuah tool yang banyak dimanfaatkan oleh para developer untuk membuat aplikasi *game*. Istilah bahasa pemrograman untuk membuat *game* disebut dengan GL (*Game Language*). Dalam Flash, *Actionscript* adalah bahasa pemrograman yang dipakai untuk mengatur dan mengolah aset.

3.7 Uji Prototype

Tahapan pengujian dilakukan agar diketahui berbagai kesalahan yang mungkin dilakukan, termasuk kesalahan pemrograman.

Tabel 1 : Pengujian memulai permainan

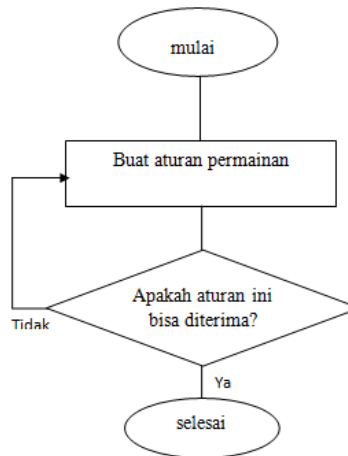
NO	Skenario pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Memilih option main pada menu utama untuk memulai permainan	Arahkan kursor pada tombol, klik kiri pada mouse	Tampilan berpindah ke frame cerita pembuka.	Tampilan berpindah ke frame cerita pembuka.	Valid
2	Memilih option main pada menu utama untuk memulai permainan	Arahan kursor tidak pas pada tombol main, klik kiri pada mouse.	Tampilan berpindah ke frame cerita pembuka.	Tampilan tidak berpindah ke frame cerita pembuka	Tidak valid
3	Memilih option main pada menu utama untuk memulai permainan	Arahkan kursor pada tombol main, klik kanan pada mouse	Tampilan berpindah ke frame cerita pembuka	Tampilan tidak berpindah ke frame cerita pembuka	Tidak valid

Tabel 2 : Pengujian tembak target

NO	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menembak di area tembak atau di sekitar <i>enviromtent</i> (lingkungan) dalam level	Arahan <i>crosshair</i> tidak tepat pada target, klik kiri pada mouse	Ada suara tembakan, Peluru pada indikator berkurang, penambahan skor	Ada suara tembakan, Peluru pada indikator berkurang, skor tidak bertambah	Tidak Valid
2	Menembak di area tembak atau di sekitar <i>environment</i> (lingkungan) dalam level	Arahkan <i>crosshair</i> tepat pada target, klik kiri pada mouse	Ada suara tembakan, Peluru pada indikator berkurang, tambah skor dan isi bar target, animasi target hancur.	Ada suara tembakan, Peluru pada indikator berkurang, skor dan isi bar target bertambah, animasi target.	Valid
3	Menembak di area tembak atau di sekitar <i>environment</i> (lingkungan) dalam level	Tekan tombol karakter huruf dan angka pada keyboard	Ada suara tembakan, Peluru pada indikator berkurang, penambahan skor	Tidak ada suara tembakan, Peluru pada indikator tidak berkurang, tidak ada penambahan skor	Tidak valid

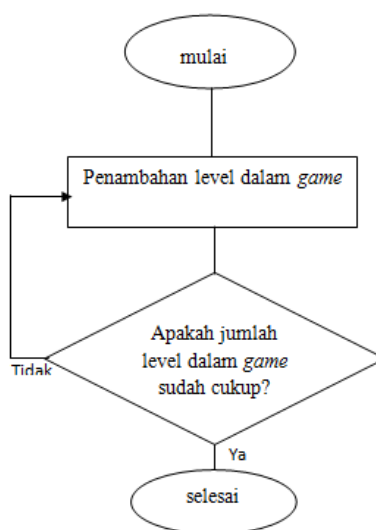
3.8 *Prototype User's System Diagram*

Prototype User's Diagram System adalah istilah dalam *prototyping* untuk sebuah diagram yang digunakan untuk mengidentifikasi bagian-bagian dari perangkat lunak yang di-*prototype*-kan. *Prototype User's diagram system* digambarkan dengan menggunakan flowchart, dan dibuat sesuai dengan beberapa hal yang diinginkan oleh *user*.



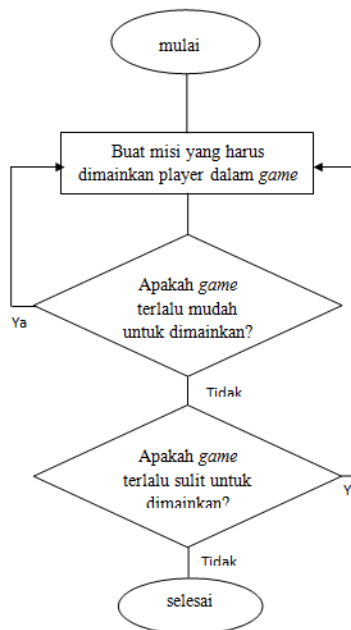
Gambar 12 : Flowchart aturan main dalam *prototype game*

Diagram diatas menggambarkan pembuatan aturan permainan dalam *prototype game* yang telah dibuat. Penambahan variasi aturan permainan merupakan bagian dari *prototype* yang akan disosialisasikan kepada *user*, apakah aturan tersebut bisa diterima oleh kebanyakan *player* atau tidak. Jika direspon dengan baik, maka aturan tersebut akan ditetapkan sebagai bagian dari permainan. Namun jika tidak, maka harus dibuat aturan baru.



Gambar 13 : Flowchart penambahan level dalam *prototype Games*

Diagram ini menggambarkan level yang sudah ditambahkan ke dalam *prototype game*, apakah jumlah level dalam *game* sudah cukup atau tidak. Sebab level yang terlalu sedikit akan membuat pemain kurang puas sementara level yang terlalu banyak bisa jadi membuat pemain bosan, oleh karena itu jumlah level harus disesuaikan dengan kebutuhan. Jika jumlah level sudah cukup, maka tidak perlu ada lagi penambahan, namun jika masih kurang maka perlu adanya penambahan level lagi.



Gambar 14 : Flowchart pembuatan misi dalam *prototype Games*

Misi dalam setiap level juga harus diperhatikan. Apakah misi tersebut terlalu mudah untu dimainkan atau justru permainan menjadi terlalu su lit untuk dimainkan. Jika terlalu mudah, maka harus dibuat kembali rancangan misi yang lebih sulit. Jika permainan terlalu sulit untuk dimainkan, maka harus dikurangi tingkat kesulitan misinya.

3.9 Evaluasi dengan User

Tahap selanjutnya ialah mengevaluasi hasil prototype dengan user. Pada tahapan ini juga dilakukan pengujian oleh pemain dari berbagai kalangan agar didapatkan masukan dari berbagai sudut pandang yang berberbeda. Hal tersebut berguna untuk perbaikan aplikasi *game*.

3.10 Transformasi *Prototype* ke Sistem Penuh

Langkah terakhir dalam perancangan aplikasi *game* adalah menstranformasikan *prototype* yang telah selesai dibuat menjadi sistem penuh. Dalam kata lain, ini adalah tahap penyempurnaan sebuah sistem *game*. Tahap ini dilakukan dengan penghilangan beberapa bagian yang tidak dibutuhkan, atau penambahan bagian yang diperlukan, serta melakukan berbagai perbaikan lain agar *game* dapat berjalan sesuai dengan yang diharapkan.

4. KESIMPULAN

Berdasarkan atas apa yang telah dijelaskan pada bab-bab yang lalu serta berdasarkan atas jawaban dari masalah yang telah teridentifikasi sebelumnya, maka didapatkan kesimpulan sebagai berikut :

1. Ilmu komputer sangat penting untuk dipelajari karena ia dapat diterapkan dalam berbagai bidang. Salah satunya adalah dalam bidang *entertainment* dan atau dalam bidang bisnis, dimana penerapan ilmu komputer dapat menghasilkan sebuah produk berupa aplikasi *game* untuk komputer. Tujuan umum dibuatnya *game* adalah sebagai media yang dapat menghibur penggunanya dan merubah suasana menjadi lebih menyenangkan. Selain itu mengingat peminatnya yang banyak, maka menjadi seorang *game developer* dapat menjadi pilihan karir karena dalam dunia *game* terdapat peluang bisnis yang memiliki prospek yang cukup bagus.
2. Pada level satu dalam *game first person shooting* ini, pengaturan jarak antar objek tidak bergerak yang dijadikan sebagai target dimaksudkan untuk melatih fokus pemain, terutama pada saat pemain diharuskan untuk memusatkan pandangan pada target yang terlihat kecil karena letaknya paling jauh. Level 2 dalam *game* ini juga dibuat bagi pemain agar fokus pada target utama, apalagi tidak semua target terlihat, apalagi tidak semua target terlihat tetapi ada beberapa yang tersembunyi. Seperti yang terdapat pada level 3 dan 4, menembak objek yang bergerak adalah salah satu latihan sederhana untuk daya tanggap pemain. Level 5 dan 6 dirancang untuk melatih daya ingat, dimana konsep dari kedua level ini adalah target yang tersembunyi dengan menyisipkan objek terlarang yang tidak boleh ditembak diantara tempat persembunyian target utama. Pada

level 5 tidak ada batasan waktu agar pemain dapat mempelajari pola keberadaan target lebih lama, dan ketika maju ke level 6 pemain dituntut untuk mengingat pola tersebut dan bergerak dengan cepat karena adanya keterbatasan waktu. Level terakhir menitik beratkan pada fokus karena keterbatasan area tembak, dimana pemain harus terus mengikuti arah gerak lingkaran sambil mencari target yang hanya dapat terlihat di area dalam lingkaran tersebut. Dari rancangan keseluruhan level dalam game tersebut maka dapat disimpulkan bahwa, seperti halnya konsep dalam game bertema First Person Shooting pada umumnya, game yang dirancang juga memiliki konsep *focus*, *responsiveness*, dan *memory capability*. Dapat disimpulkan pula bahwa First Person Shooting bukan hanya sekedar permainan menembak saja. Tetapi secara tidak langsung dapat digunakan sebagai media latihan ringan untuk otak. Hal ini dapat menjadi salah satu sisi positif dalam bermain *game*.

3. Game sederhana dengan genre First Person Shooting ini memiliki tingkat kesulitan yang berbeda dalam setiap levelnya, tetapi alur permainannya mudah untuk dipelajari dan dimainkan oleh siapapun sehingga *game* ini dapat masuk ke semua kalangan masyarakat. Selain itu di dalam *game* ini tidak ada unsur kekerasan tetapi lebih cenderung seperti permainan olahraga menembak, sehingga setidaknya dapat merubah penilaian masyarakat yang memandang *game* bertema menembak tidak selalu membawa pengaruh positif.

5. DAFTAR PUSTAKA

- Crawford, Chris. 1997. *The Art of Computer Game Design*. Washington : Washington State University.
- Crawford, Chris. June 2003. *Chris Crawford on Game Design*. United States of America : New Riders Publishing
- Hidayatullah, Priyanto, S.T, Akbar, M. Amarullah, Rahim, Zaky, Februari 2011, *Animasi Pendidikan menggunakan Flash*, Bandung : Informatika
- Hidayatullah, Priyanto, S.T, Daswanto, Andi, Nugroho, Sulistyono, September 2011, *Membuat Mobile Game Edukatif Dengan Flash*, Bandung : Informatika
- Hutchinson, Haigh. 2009. *Real Time Cameras : A guide for Game Designers and Developers*. New York : Morgan Kauffman Publisher.

- Indrajani, S.Kom, MM dan Martin, S.Kom., 2004. *Pemrograman Berorientasi Objek dengan Java*, Jakarta :Elex Media Komputindo.
- Rosenzweig, Gary. 2008. *Actionscript 3.0 Game Programming University*. Indiana : Que.
- Rosenzweig , Gary. January 2011. *Actionscript 3.0 Game Programming University 2nd Edition*. Indiana : Que.
- Sridadi, Bambang, Ir.,M.Sc. , Maret 2010 , *Sistem Waktu Nyata (Real-time System)*, Bandung : Informatika
- Schell, Jesse. 2008. *The Art of Game Design*. Oxford : Morgan Kauffman Publisher.
- Van Der Spuy, Rex. 2010. *Advanced Game Design with Flash*. New York : Friends of.
- Wibawanto, Wandah . 2013. *Memprogram Game Flash 3D itu mudah*, Yogyakarta : Andi Offset.
- Wibawanto, Wandah. 2006. *Dasar – dasar Pemrograman Flash Game*. Yogyakarta : Andi Offset.