**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

1. **IDENTITAS PENGUSUL**

Nama : **RADIK KRIOLAMPAH**

NRP : **5107100149**

Dosen Wali : **DWI SUNARYONO, S.Kom, M.Kom.**

1. **JUDUL TUGAS AKHIR**

Rancang Bangun *Editor* pada *Extensible Flash Game*

1. **LATAR BELAKANG**

Industri video game atau sering diistilahkan dengan *interractive entertainment* adalah salah satu sektor ekonomi meliputi pemasaran dan pengembangan game. Industri video game mencakup berbagai disiplin ilmu dan saat ini telah mampu mempekerjakan ribuan orang di seluruh dunia.

Perkembangan game semakin hari semakin pesat. Banyak game-game fenomenal seperti Final Fantasy, Halo, Warcraft, dan sebagainya yang memiliki kompleksitas sangat tinggi baik dari segi grafis, *gameplay*, maupun *story* tidak henti-hentinya meluncurkan sekuel dari *release* sebelumnya. Belum lagi game-game yang berjalan di *handphone*, PSP, XBox, Nitendo Wii, dan Playstation yang hampir tak terhitung jumlahnya dan bisa dibayangkan berapa dollar perputaran uang yang terjadi dalam bisnis *interractive entertainment* ini.

Namun kebanyakan dari game tersebut bersifat *corporate*, hasil karya perusahaan yang dikerjakan oleh tim profesional mulai dari perancangan, implementasi sampai dengan pemasaran, sehingga tampaknya membuat game dengan proses bisnis semacam itu sangat sulit apabila dikerjakan secara *indie* tanpa bergabung dengan perusahaan tertentu.

Untuk itu, salah satu cara paling mudah untuk membuat game yang mampu dipublikasikan untuk tujuan komersil tanpa harus “ribet” adalah membuat sebuah game yang bersifat *free-to-play* namun memiliki celah untuk mendapat pemasukan seperti *item mall* atau *sponsorship*. Dan salah satu *genre game* yang paling mungkin dibuat untuk melakukan hal di atas adalah *web-based / browser game*.

Bisa dimainkan tanpa melakukan proses instalasi, bisa diakses dari mana saja menjadi keunggulan tersendiri bagi *browser game*. Game-game seperti Steambird, Farmville, Travian, dan *browser game* lainnya telah membuktikan bahwa game-game berbasis web juga bisa meraup keuntungan.

Pembuatan *browser game* sebenarnya dapat dilakukan hanya dengan menggunakan *html, css,* dan *javascript*. Namun akan lebih mudah dan kompleks apabila dibuat menggunakan bahasa lain yang mampu dijalankan *browser* *internet* meskipun membutuhkan sebuah plugin. Dan plug-in yang paling banyak digunakan pada *browser internet* saat ini adalah plug-in Flash.

*Flash game* merupakan salah satu jenis *browser game* dengan perkembangan sangat pesat khususnya dalam hal inovasi. Hampir setiap bulan muncul *Flash game* dengan sistem permainan baru yang merupakan hasil pemikiran kreatif dari *game* yang ada sebelumnya.

Kemudahan dalam pemasaran *Flash game* menjadi salah satu faktor yang membuat para *developer* bersaing untuk membuat *Flash game* yang interaktif dan menarik. Selain itu industri di bidang *browser game* juga menjadi salah satu jenis industri yang berkembang dengan pesat, salah satunya adalah Zynga dengan Farmville-nya yang mampu menyedot 33 juta pemain dalam waktu dua bulan.

Adobe Flash, (dahulu bernama Macromedia Flash) adalah sebuah multimedia *platform* untuk menambahkan animasi, video dan interaktivitas pada halaman web. Namun seiring dengan perkembangan teknologi Flash, Flash juga digunakan untuk membuat *browser game*. Dari banyaknya jenis *browser game* yang ada, *browser game* dengan *platform* Flash memegang *market share* terbesar, hal ini dapat diketahui dari *plugin* flash yang menjadi *plugin* yang paling banyak terinstall di *browser* secara global.

Namun di sisi lain, kemudahan pengembangan dan pemasaran *Flash game* menimbulkan masalah baru, developer dan *Flash game* yang bermunculan menyebabkan tingkat persaingan yang semakin tinggi. Akhir-akhir ini penjualan *Flash game-*pun menurun karena banyaknya *Flash game* yang muncul namun dengan *gameplay* yang hampir sama dengan *Flash game* sebelumnya.

Selain itu dari segi pengembangan, mengembangkan *Flash game* yang sudah ada lebih sulit daripada membuat *Flash game* yang baru karena kebanyakan *Flash game* hanya terdiri dari sebuah *file* .swf tanpa didukung *file* eksternal sehingga proses *upgrade* harus dilakukan dengan meng-*compile* ulang program. Belum lagi ketika developer menggunakan actionscript 2 yang belum berbasis objek, kedua proses tersebut akan menjadi lebih sulit.

Dari beberapa kendala itulah penyusun mencoba untuk membuat sebuah *Flash game* yang kompleks namun mudah untuk dikembangkan. Berbeda dengan *Flash game* yang ada sebelumnya namun tetap memiliki *Extensibility* yang tinggi.

*Extensibility* adalah sebuah prinsip desain sistem yang menitikberatkan pada pengembangan di masa depan. Konsepnya adalah bagaimana membuat sebuah desain sistem yang mudah untuk dirubah namun tidak mengganggu jalannya sistem yang sudah ada.

Sedangkan *Extensible Flash game* adalah istilah yang penyusun gunakan untuk menggambarkan sebuah *Flash game* yang mampu dirubah tanpa perlu meng-*compile* ulang *Flash game*. Memiliki konsep seperti *plug-in* pada *browser internet*, dimana perubahan yang dilakukan terhadap *Flash game* tidak bersifat permanen dan mampu dibongkar pasang tanpa mengganggu program utama.

1. **RUMUSAN MASALAH**
2. Bagaimana membuat rancangan game dengan tepat untuk membuat sebuah *Flash game* yang *extensible*?
3. Bagaimana menyesuaikan rancangan game agar sesuai dengan *gameplay* yang akan diimplementasikan?
4. Bagaimana membuat sebuah *editor* yang mampu mengelola data eksternal?
5. **BATASAN MASALAH**
6. Layar utama dalam game ada enam, yaitu menu utama, cerita, peta, petualangan, perang dan pelabuhan. Susunan layar tidak bisa ditambahi atau dikurangi.
7. HUD (Head-Up Display) adalah sebuah kumpulan dari elemen pada *user interface* yang menunjukkan status permainan. Analogi dalam dunia nyata adalah panel kontrol pada pesawat terbang atau *dashboard* pada kendaraan bermotor. Untuk perubahan pada HUD, editor hanya mampu merubah skin dan sound. Letak dan susunan HUD sudah ditentukan dan tidak bisa dirubah lagi.

Contoh objek-objek HUD dalam tugas akhir ini adalah :

* Penunjuk status seperti *health*, *ability*, *item*, uang, dsb.
* Minimap.
* Cursor.
* Menu

1. Objek non-HUD dalam game adalah objek-objek yang terdapat dalam game selain HUD. Objek non-HUD ini masih dibagi lagi menjadi dua, yaitu *Playable Character* dan *Non-Playable Character (NPC)*.

* *Playable Character*, adalah semua objek yang terdapat dalam game dan dapat dimainkan oleh player. Contohnya adalah pasukan, bangunan, karakter yang dipilih dalam game *First Person Shooter (FPS)*, yang sebagian besar *behavior* atau gerakannya dapat diatur oleh player.
* Sedangkan *Non-Playable Character (NPC)*, adalah objek-objek yang terdapat di dalam game namun tidak bisa dimainkan oleh player. Contoh NPC dalam game ini adalah.

1. *Buff,* adalah animasi yang melekat pada sebuah *playable character* sekaligus menambahkan status tertentu pada *playable character* tersebut.
2. *Effect,* animasi yang hanya sesekali muncul dalam gameplay dan tidak bergerak. Contohnya adalah ledakanyang bergerak sendiri dalam gameplay (awan, pulau, peluru, dan sebagainya).

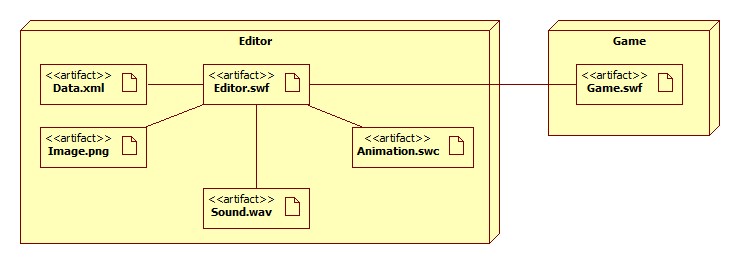
Dalam tugas akhir ini, elemen dari objek non-HUD yang bisa dirubah adalah animasi, suara, dan nilai atribut.

1. **TUJUAN PEMBUATAN TUGAS AKHIR**

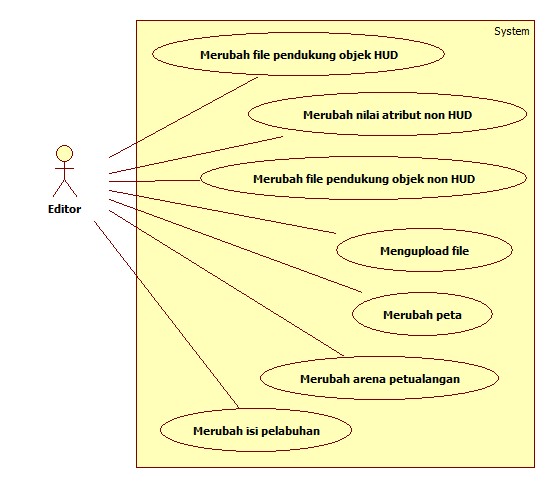
Membuat sebuah *editor* pada *extensible Flash game* sehingga mampu menghasilkan sebuah *Flash game* yang mudah untuk dikembangkan tanpa mengganggu bagian program yang sudah ada sebelumnya.

1. **RINGKASAN TUGAS AKHIR**

Pengerjaan tugas akhir ini terbagi dalam dua program besar yaitu *game* dan *editor* dimana fungsional kedua program tersebut berbeda namun saling terkait. Kedua program dipisahkan agar player tidak bisa sembarangan mengedit data-data pada game dan editor tidak bisa langsung menjalankan program yang diedit. Hubungan antara kedua program tersebut digambarkan sebagai berikut.



*Editor* adalah program yang mampu merubah hampir semua aspek yang ada di dalam *Flash* game. Hal-hal yang mampu dilakukan oleh editor antara lain :



1. Mengubah file pendukung objek HUD

File pendukung yang dimaksud adalah file image berekstensi .png untuk merubah *skin* dari HUD tanpa mengubah tata letak dan susunan dari HUD.

1. Merubah nilai atribut objek non-HUD

Seluruh objek non-HUD memiliki atribut. Berbeda dengan objek HUD, objek non-HUD tidak hanya mementingkan tampilan namun juga nilai dari atribut-atribut yang terdapat di dalamnya.

1. Merubah file pendukung objek non-HUD

Objek non-HUD memiliki file pendukung berupa suara dan animasi yang keduanya bisa dirubah melalui editor.

1. Mengupload file

Perubahan terhadap file pendukung objek tidak hanya bisa dilakukan dengan file yang sudah ada, namun juga bisa dilakukan dengan file-file baru dengan cara mengupload terlebih dahulu.

1. Merubah peta

Layar peta terdiri dari stage dan peta itu sendiri, dimana keduanya dapat dirubah melalui editor.

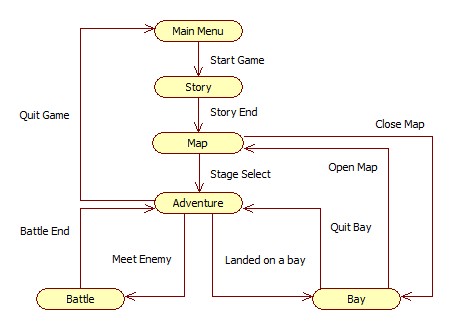
1. Merubah arena petualangan

Petualangan terjadi di permukaan laut, dimana letak pulau, musuh, harta karun bisa dirubah melalui editor.

1. Merubah isi pelabuhan

Isi yang dimaksud adalah *playable character* yang dijual di pelabuhan, ibarat toko. Editor dapat melakukan perubahan terhadap daftar *playable character* yang dijual di setiap pelabuhan.

Sedangkan game adalah program yang berjalan sesuai dengan objek-objek yang telah dimanipulasi oleh *editor*. Gambaran alur game secara umum dapat dilihat pada *sketchflow* di bawah ini.



1. Main Menu

Halaman yang pertama muncul saat memulai permainan. Dari halaman ini terdapat menu-menu utama seperti :

* Memulai permainan baru
* Melanjutkan permainan
* Melakukan konfigurasi / option
* Membuka help

1. Story

Adalah halaman berisi animasi untuk menampilkan cerita. Cerita yang dipaparkan diambil dari data eksternal sehingga mudah untuk dirubah.

1. Map

Layar dimana terdapat peta permainan secara keseluruhan dan stage-stage yang bisa dipilih pemain.

1. Adventure

Setelah memilih stage di halaman map. Player akan dibawa ke arena petualangan di permukaan laut. Di arena petualangan akan terdapat musuh, pulau, harta karun dan lain sebagainya.

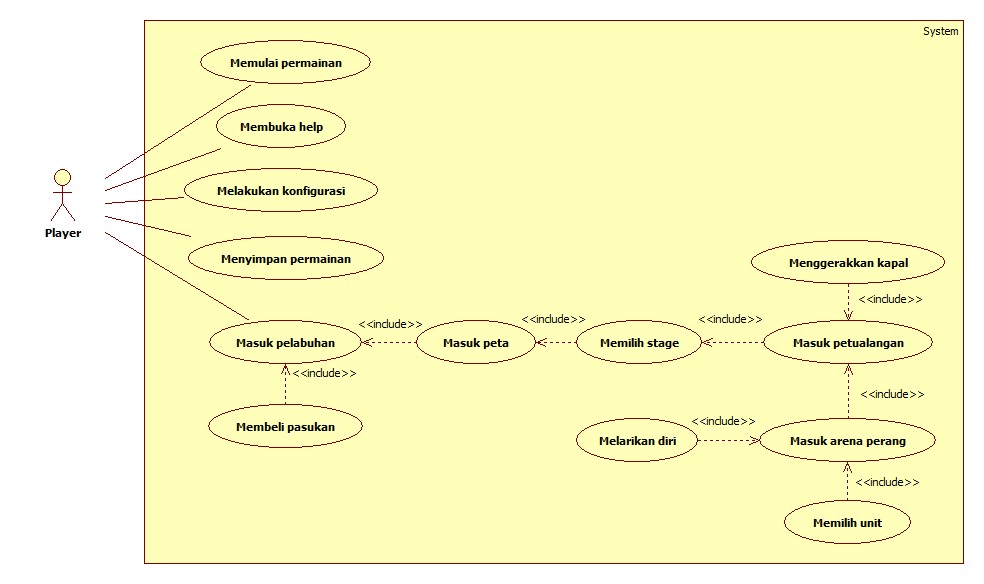
1. Battle

Ketika bertemu dengan musuh maka peperangan kemungkinan akan terjadi. Peperangan bisa dimulai baik oleh musuh atau player itu sendiri.

1. Bay

Quest / misi dalam game hanya dapat diselesaikan dengan cara mencapai pelabuhan. Selain itu, proses jual beli juga dilakukan di pelabuhan.

Kemudian hal-hal yang mampu dilakukan player dalam game terangkum dalam use case di bawah ini.



1. **TINJAUAN PUSTAKA DAN DASAR TEORI**
2. **Extensibility**

Ciri-ciri software yang baik antara lain memiliki *minimal complexity, ease of maintenance, loose coupling, extensible, reusable, high fan-in, low to medium fan-out, portable, leanness,* dan *using standard techniques.*

*Flash game* yang dibangun menggunakan actionscript 3 sudah bisa menerapkan hampir kesemua aspek tersebut, namun sayang kebanyakan *Flash game* masih memiliki kelemahan pada e*xtensibility* karena kebanyakan *Flash game* hanya terdiri dari sebuah *file* .swf tanpa didukung *file*-*file* eksternal. Perubahannya pun harus dilakukan dengan *recompile* *file* .fla karena source-source pada *Flash game* tidak bisa dirubah tanpa membongkar terlebih dahulu *file* .fla.

*Extensibility* yang diterapkan pada tugas akhir ini mampu melakukan perubahan pada *Flash game* mulai dari gambar sampai nilai atribut setiap objek tanpa melakukan *recompile file*. fla. Karena perubahan yang dilakukan tidak permanen, maka *editor* pada game ini hanya merubah *file-file* eksternal. Gambaran e*xtensibility* ini hampir sama dengan konsep *plug-in* pada *browser internet* yang mampu dibongkar pasang tanpa mengganggu program yang sudah ada.

1. **Portable Network Graphics (PNG)**

*File* png mampu menghandle transparency dan memiliki ukuran kecil. Karena itulah editor pada tugas akhir ini hanya menghandle .png untuk *file* gambar karena telah banyaknya editor *file* .png dan compatibility *file* .png yang tinggi.

1. **SWC**

Format animasi yang dapat dihubungkan dengan *file* .swf. Sementara ini editor untuk *file* .swc adalah Adobe Flash CS4 ke atas dan Adobe Flex. Kelebihan dari *file* .swc adalah kemampuannya untuk dikontrol dari dalam program. Berbeda dengan *file* .swf yang bisa dikatakan sudah “jadi”, frame-frame dalam *file* .swc masih bisa digerakkan menggunakan *code*.

1. **WAV**

Editor pada tugas akhir ini hanya menerima input suara berupa *file* .wav. *File* .wav meskipun memiliki ukuran besar dan kualitas standar, memiliki compatibility yang tinggi dengan *platform* Flash. Memang banyak format suara lain yang lebih efisien seperti *file* .mp3, namun terkadang penggunaannya pada platform Flash masih bermasalah.

1. **ActionScript 3**

Action Script adalah bahasa pemrograman yang dibuat oleh Macromedia Inc (sekarang dimiliki oleh adobe) dan digunakan untuk mengembangkan *website* dan *software* pada *platform* Adobe Flash *player*, dan digunakan di halaman web dalam bentuk *file* swf yang *embeded*.

Flash MX 2004, sebuah *scripting* *programming* *language* yang lebih cocok untuk mengembangkan aplikasi flash, seringkali dalam sebuah kasus lebih menghemat tenaga dengan *scripting* daripada membuat animasinya dengan manual. Setelah kemunculan ActionScript 2 mulai bermunculan game berbasis *platform* flash.

Sejak kemunculan Flash Player 9 (2006) versi baru dari ActionScript dirilis, ActionScript 3. Berbeda dari sebelumnya, ActionScript 3 adalah *Object Oriented* *Programming* (OOP) sehingga developer lebih mempunyai kontrol dan code *reusability* ketika membangun sebuah aplikasi Flash yang kompleks. ActionScript 3 ditujukan untuk versi Flash Pemain 9 dan lebih tinggi. Performa ActionScript 3 lebih cepat 10 kali daripada action script versi sebelumnya.

1. **Library Box2D**

Box2D adalah sebuah library untuk membantu mensimulasikan hukum fisika ke dalam sebuah aplikasi. Box2D pertama kali di presentasikan oleh Erin Catto di [*Game Developers Conference*](http://en.wikipedia.org/wiki/Game_Developers_Conference) (GDC 2006). Pada 11 September 2007 di rilis sebagai *open source* di *Sourceforge*. Pada 17 Januari 2010 Box2D berpindah ke Google *code* untuk *hosting*.

Box2D pada mulanya dirilis dalam bahasa C++ dan telah digunakan pada Nintendo DS, Wii dan telepon genggam (termasuk Android dan Iphone). Sekarang Box2D telah di porting ke berbagai macam bahasa pemrograman lain termasuk Adobe Flash (ActionScript dan haxe), Java, C# dan JavaScript.

Box2D dapat mensimulasikan suatu benda yang terdiri dari *convex, polygon, circles* dan *edge shapes* (siku). Benda dapat dihubungkan dengan *joints* (sendi). Box2D memungkinkan simulasi hukum fisika dari banyak benda yang bergerak cepat bertumbukan dengan tanpa banyak mengurangi *performance* komputer.

Dalam game ini box2D digunakan untuk mendeteksi tabrakan yang terjadi selama mode petualangan. Karena player nanti akan menghadapi berbagai *obstacle* yang membutuhkan efek fisika seperti saat bertabrakan.

1. **EXtensible Markup *Language* (XML)**

Adalah *file* yang digunakan editor untuk menyimpan data dan digunakan *Flash game* sebagai panduan untuk membaca *file* eksternal. Penyusun menggunakan XML untuk menampung data daru objek-objek game karena Flash masih belum mendukung sepenuhnya penggunaan DBMS.

1. **METODOLOGI**
2. Analisis Kebutuhan dan Studi Literatur

Untuk menghindari pengerjaan tugas akhir “macet” di tengah jalan, pada tahap awal penyusun akan mencari sebanyak-banyaknya literatur yang mendukung sehingga tidak mengulang kesalahan yang sudah pernah dilakukan oleh developer lain sehingga tugas akhir dapat diselesaikan dengan lebih cepat. Literatur yang digunakan adalah terkait dengan library *box2D, actionscript 3, xml*, dan tentunya *adobe flash* itu sendiri.

1. Perancangan Sistem

Kesuksesan *software* berbasis OOP tentunya sangat bergantung pada rancangan sistem. Untuk itu penyusun menghabiskan waktu lebih untuk membuat rancangan awal sistem, sehingga ke depannya implementasi *Flash game* game dapat dilakukan tanpa perlu khawatir dengan desain awalnya.

1. Implementasi

Pada tahap ini dilakukan pembuatan perangkat lunak yang merupakan implementasi dari rancangan yang telah dibuat sebelumnya.

1. Uji coba dan evaluasi

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak menggunakan data atau skenario yang telah dipersiapkan sebelumnya. Uji coba dan evaluasi perangkat dilakukan untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya program, dan mengadakan perbaikan jika ada kekurangan.

1. Penyusunan Buku Tugas Akhir

Pada tahap ini melakukan pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir. Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

Secara garis besar, buku tugas akhir nantinya terdiri atas beberapa bagian yaitu :

1. Pendahuluan
   1. Latar Belakang
   2. Permasalahan
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Uji Coba dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka
7. **JADWAL KEGIATAN**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Tahap | Mei | | | | Juni | | | | Juli | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
|  | Analisa kebutuhan dan studi literatur |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Uji coba dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |

1. **DAFTAR PUSTAKA**
2. Wikipedia, 2011, *Browser Game*, [online],

(<http://en.wikipedia.org/wiki/Browser_game> diakses tanggal 1 Mei 2011)

1. FlashGameLicense, 2011, *Sales Statistics*, [online],

(<http://www.flashgamelicense.com/report_monthly_site_sales.php>

diakses tanggal 1 Mei 2011)

1. Wilson, Greg, 2006, *Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design,* [online], <http://www.gamasutra.com/view/feature/2538/off_with_their_huds_rethinking.php> diakses tanggal 11 Mei 2011)
2. Shawn, 2009, *Characteristics of Good Program Design*, [online],

<http://www.devirtuoso.com/2009/06/10-characteristics-of-good-program-design/> diakses tanggal 11 Mei 2011)

1. Wikipedia, 2011, *HUD (video gaming)*, [online],

<http://en.wikipedia.org/wiki/HUD_%28video_gaming%29>

diakses tanggal 11 Mei 2011)

1. Wikipedia, 2011, *Extensibility*, [online],

http://en.wikipedia.org/wiki/Extensibility

diakses tanggal 13 Mei 2011)

1. Nagle, Robert, 2001, *Learning with "Extensible Game Systems",* [online], <http://www.imaginaryplanet.net/essays/literary/games/games3.php>

diakses tanggal 13 Mei 2011)