

Studi tentang Metodologi Pengembangan Game untuk Program Studi Teknik Multimedia Jaringan di Politeknik Negeri Batam

Riwinoto* , Liony Lumombo* , Selly Artaty Zega* , Cahya Miranto*

* Politeknik Negeri Batam

Jurusan Teknik Informatika, Program Studi Teknik Multimedia Jaringan

Parkway Street, Batam Centre, Batam 29461, Indonesia

E-mail: riwi@polibatam.ac.id, liony@polibatam.ac.id, selly@polibatam.ac.id, cahya@polibatam.ac.id

Abstrak

Abstrak: Game mempunyai dua karakteristik yaitu sebagai software dan sebagai konten multimedia. Kedua karakteristik ini menyebabkan pengembangan game harus mempunyai metodologi yang melingkupi kedua karakter tersebut. Silabus pemrograman game di Program studi Multimedia Jaringan menyatakan bahwa pengembangan game didukung oleh banyak faktor seperti naskah produksi, desain grafis, animasi, pemrograman, basis data dan intelegensi buatan. Untuk mempermudah pemahaman mahasiswa, dosen pembimbing dan dosen penguji maka dibuatlah metodologi pengembangan game yang menjadi standar dokumen bagi mahasiswa Tugas Akhir atau Proyek Akhir dengan tema game. Metodologi yang digunakan adalah gabungan dari metodologi pengembangan software dan pengembangan konten multimedia. Metodologi tersebut menghasilkan dokumen yaitu spesifikasi kebutuhan perangkat lunak, diagram use case yang diperkaya dengan diagram *game flow*, diagram *robustness*, diagram *sekuens*, diagram aktivitas, diagram kelas, *scriptwriting* dan *storyboard*.

Kata kunci: Game, software, konten multimedia, diagram use case, diagram *game flow*, *scriptwriting* dan *storyboard*

Abstract: Games have two characteristics, namely as software and as multimedia content. Both of these characteristics led to the development of the game must have a methodology that encompasses both the characters. Syllabus game programming in Network Multimedia study program stated that game development is supported by many factors such as production script, graphic design, animation, programming, databases and artificial intelligence. To facilitate the understanding of the student, the supervisor and the examiner then made the game's development methodology is the standard document for students final project or final project with the theme of the game. The methodology used is a combination of software development methodologies and the development of multimedia content. This methodology resulted in a document that is the software requirements specification, use case diagrams enriched with game flow diagrams, robustness diagrams, sequence diagrams, activity diagrams, class diagrams, scriptwriting and storyboarding.

Keywords: Games, software, multimedia content, use case diagrams, flow charts games, scriptwriting and storyboarding

1 Pendahuluan

Pengembangan game menjadi hal yang penting karena game mempunyai karakteristik unik sebagai salah satu tipe dari software. Karakteristik tersebut adalah pertama game memiliki aspek multimedia seperti musik, grafik, seni dan cerita seperti dijelaskan oleh [1].

Karakteristik kedua adalah skala pengembangan game bisa berkembang lebih besar, cepat dan kompleks sesuai dengan keinginan pengguna game [2]. Akibatnya pengembang game mengalami kesulitan dalam komunikasi antara pihak pengembangan seperti programmer dengan artis yang memodelkan karakter, analisis software dengan. Pengembangan software pada umumnya menggunakan pendekatan berbasis objek [3].

Pendekatan berbasis objek mempunyai diagram UML yaitu use case, diagram *robustness*, diagram sekuens, diagram aktivitas, diagram kelas, diagram komponen, diagram deployment. Penggunaan diagram UML tersebut masih memiliki keterbatasan ketika harus diterapkan pada proses-proses yang memiliki aspek multimedia.

Referensi [4] mengusulkan format desain level game individu yang dinamakan desain *game flow* sehingga pengembangan desain game lebih formal seperti pengembangan software. Desain *game flow* memperluas penggunaan Diagram use case sehingga cocok untuk desain game komputer. Hasilnya Pendekatan ini menyediakan alur game untuk level game yang ada. Oleh karena itu Pendekatan desain *game flow* dapat digunakan oleh programmer game, artist, animator, musisi dan penguji game sebagai media komunikasi. Desain *game flow* dapat melengkapi *storyboard* game sehingga memperjelas isi *storyboard* pada bagian-bagian yang sulit dan kompleks untuk didekripsikan secara eksplisit [4].

Sisi lain dari UML adalah tidak semua project software harus menggunakan seluruh diagram UML. Referensi [5] menyatakan bahwa walaupun UML adalah *lingua franca* (standar yang diterima) dari pengembangan software namun pengembang software ada yang tidak menggunakan seluruh (penggunaan parsial) diagram UML mencapai 22% . Dari penelitian referensi [5] didapatkan laporan indikatif bahwa dari pengembangan software yang menggunakan diagram UML secara parsial, 63.4 % menggunakan diagram kelas, 54% menggunakan diagram sekuens, 54 % menggunakan diagram aktivitas, 27% menggunakan diagram transisi state dan 9% menggunakan diagram use case.

Pengembangan Game di Program Studi Teknik Multimedia, Jurusan Teknik Informatika, Politeknik Negeri Batam mempunyai ruang lingkup yang luas. Pengembangan game diimplementasikan dalam dua mata kuliah inti dan beberapa mata kuliah pendukung. Mata kuliah inti adalah di animasi 2D untuk game 2D dan pemrograman game untuk game 3D. Mata kuliah pendukung adalah naskah produksi, desain grafis, algoritma pemrograman, struktur data, basis data dan kecerdasan buatan [6]. Karena ruang lingkup yang luas, diperlukan suatu format standar dalam pengembangan game. Format tersebut diharapkan menjadi acuan bagi mahasiswa yang mengambil topik tugas akhir yang menghasilkan produk game.

Jadi dalam studi ini, penulis mengusulkan format pengembangan game yang fleksibel dengan karakteristik game sebagai software dan sebagai konten multimedia.

2 Landasan Teori

A. Diagram Unified Modelling Language (UML)

Referensi [7] menyatakan Diagram UML merupakan standar terbuka dari pengembangan software. UML adalah hasil integrasi metode pengembangan OMT, OOSE dan Booch yang dikembangkan oleh Jim Rumbaugh, Ivar Jacobson dan Grady Booch.

Diagram use case

Referensi [6] menyatakan diagram 3 banyak digunakan untuk merepresentasikan kebutuhan fungsional dari perangkat lunak dalam bentuk visual dengan objek yang terlibat adalah aktor dan use case. Use case adalah unit fungsionalitas yang disediakan oleh sistem.

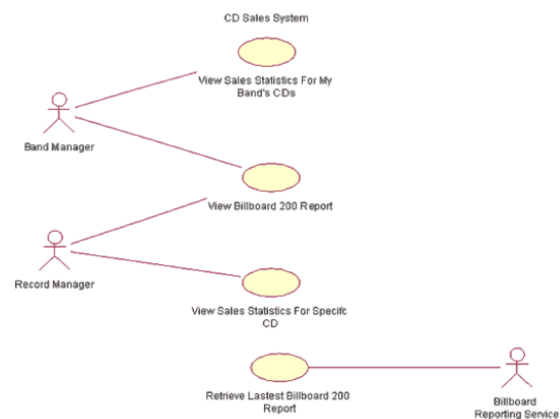


Figure 1. Contoh diagram use case

Gambar di atas merupakan contoh diagram use pada sistem delivery billboard [7].

Diagram sekuens

Diagram sekuens menggambarkan sekuensi komunikasi (message) antar objek-objek . Referensi [7] menyatakan diagram sekuens memiliki dimensi yaitu dimensi vertikal menggambarkan terjadinya komunikasi terjadi berdasarkan waktu. Sedangkan dimensi horizontal menggambarkan objek satu(yang mengirim pesan) dengan objek lain (yang menerima pesan). Referensi [6] menyatakan bahwa diagram sekuens digunakan untuk mengeksplisitkan kebutuhan software, memunculkan perilaku objek dan history penciptaan objek.

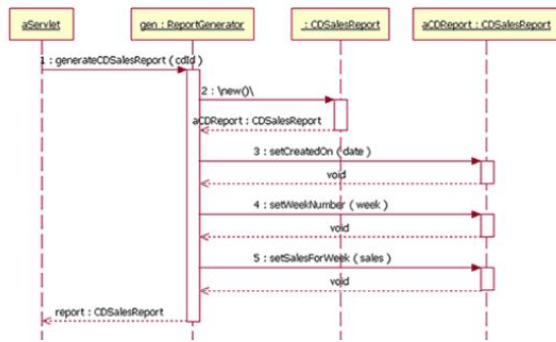


Figure 2. Contoh Diagram sekuens

Gambar di atas menggambarkan diagram sekuens untuk membuat report penjualan produk [7].

Diagram Kelas

Diagram kelas adalah diagram yang menggambarkan secara visual entitas dan relasinya terhadap entitas lain Referensi [7] menyatakan entitas bisa merupakan user, interface, maupun data.

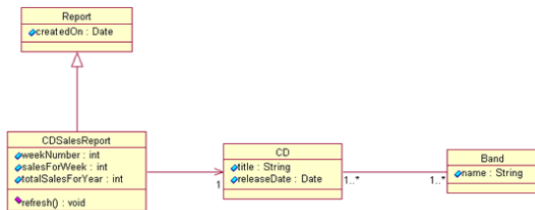


Figure 3. Contoh diagram Class

Gambar di atas merupakan contoh kelas yang menggambar entitas yang berhubungan dengan entitas (kelas) yang lain [7]. Proses identifikasi kelas meliputi 2 tahap yaitu menentukan potensial kelas dan menentukan perilaku (*behaviour*) kelas.

Diagram Aktivitas

Diagram aktivitas menggambarkan aliran prosedural dari kontrol dua atau lebih objek ketika terjadi aktivitas. Menurut referensi [7], terdapat 4 elemen dari diagram aktivitas yaitu aktivitas, asosiasi, kondisi dan *constraint*.

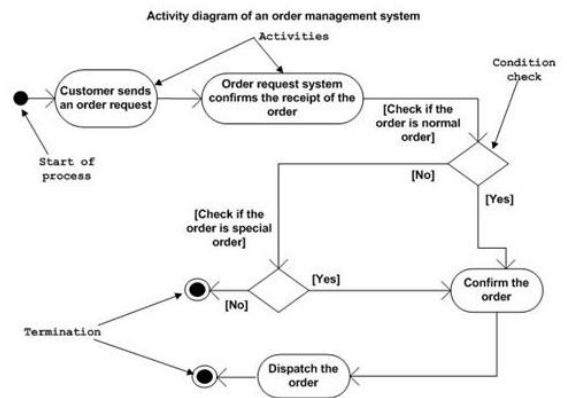


Figure 4. Contoh Diagram aktivitas

Gambar di atas menggambarkan diagram aktivitas untuk Sistem manajemen pemesanan.

Diagram Transisi State

Diagram transisi *state* memodelkan *state* yang mungkin terjadi terhadap kelas. Referensi [7] menyatakan tidak setiap kelas memiliki diagram *state*. Yang mungkin adalah kelas yang memiliki potensial *state-state* selama aktivitas.

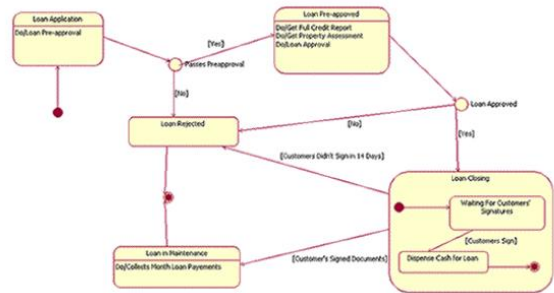


Figure 5. Contoh Diagram Transisi State

B. Diagram robustness

Ivar Jacobson memperkenalkan diagram robustness yang bertujuan untuk mengidentifikasi kelas-kelas potensial yang mungkin terjadi [7].

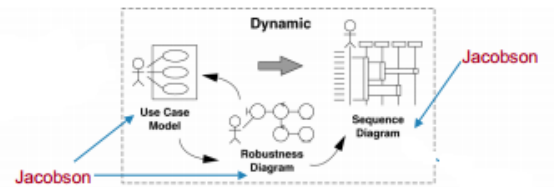


Figure 6. Posisi Diagram Robustness

Gambar diatas adalah munculnya diagram *robustness* sebagai jembatan antara diagram use case dengan diagram sekuens.

Pada diagram robustness, dikenal 3 jenis istilah yaitu

objek *boundary*, objek entitas, *controller* [7]. Objek *boundary* adalah objek interface antara sistem dengan dunia luar. Biasanya objek *boundary* adalah layar. Objek entitas adalah kelas yang bersifat informasi. *Controller* adalah relasi antara objek *boundary* dengan objek *boundary*, objek *boundary* dengan objek entitas dan objek entitas dengan objek entitas. Pada saat objek teridentifikasi maka dibuat daftar atribut dari objek tersebut. Atribut tersebut yang akan menjadi atribut pada kelas dari objek. Identifikasi nama kelas juga bisa didapatkan dari domain model dari usulan Rumbaugh. Seperti gambar berikut.

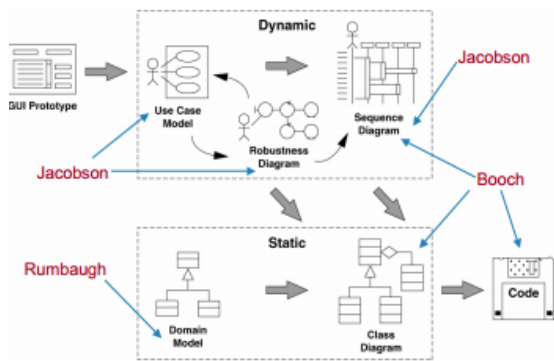


Figure 7. Model Domain

Model domain merupakan daftar kelas sementara yang hanya mempunyai atribut yang terbatas.

Berikut adalah simbol pada diagram *robustness*.

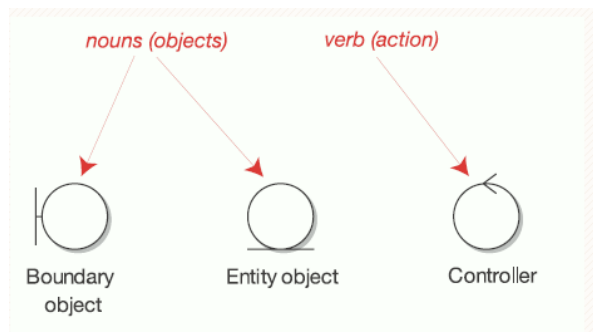


Figure 8. Simbol di Diagram Robustness

Objek *Boundary* bersifat *nouns* dan *controller* bersifat *verbs*. Beberapa aturan ketika membuat diagram *robustness* adalah *noun* dapat berkomunikasi dengan *verbs*, *noun* tidak dapat berkomunikasi dengan *noun* lain dan *verbs* dapat berkomunikasi dengan *verbs* lain.

Diagram *robustness* bukanlah diagram standar UML namun sebagai diagram alat bantu dalam mengidentifikasi diagram kelas dan sekuens.

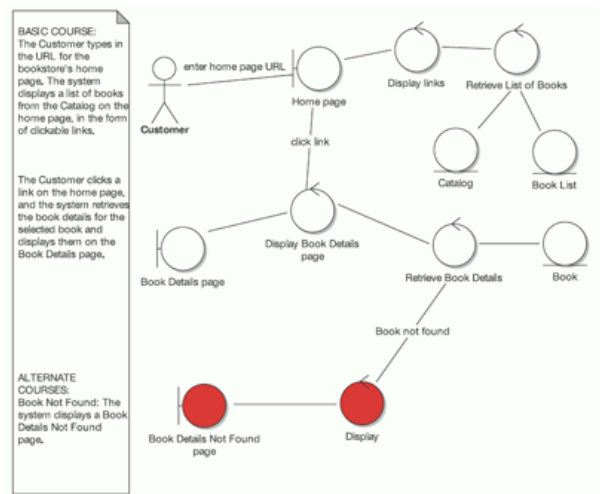


Figure 9. Contoh Diagram Robustness

Gambar adalah contoh diagram *robustness* dalam use case show book details [7].

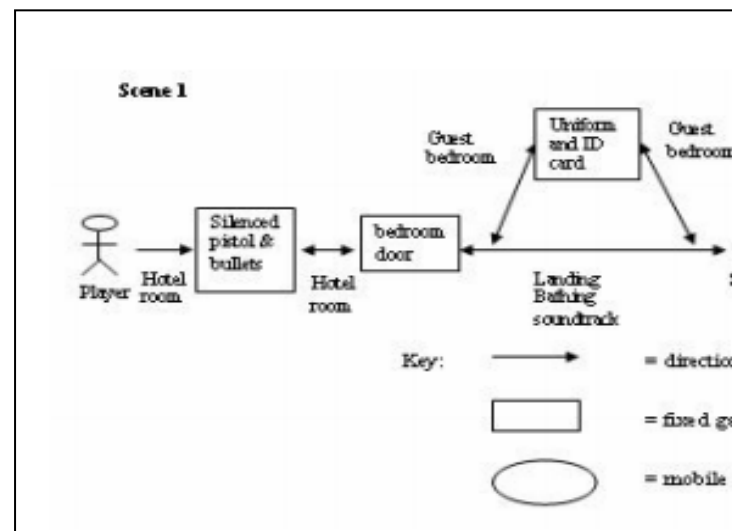


Figure 10. Contoh Game Flow Diagram

D. Scriptwriting

Scriptwriting adalah penulisan dokumen yang menjelaskan persepsi pendengaran, visual, tingkah laku dan elemen bahasa yang dibutuhkan untuk menjelaskan sebuah cerita [8]. *Scriptwriting* yang digunakan dalam pembuatan game sering disebut dengan *gamescripting* [9]. Secara umum *scriptwriting* untuk game tidak berbeda secara substansi dengan *scriptwriting* konten multimedia yang lain seperti film animasi. Perbedaannya adalah adanya interaktivitas dengan user pada *scriptwriting* game.

C. Diagram Desain *Game Flow*

Diagram Desain *Game Flow* dikembangkan oleh Taylor[4]. Tujuan dari diagram desain game flow adalah menyediakan teknik desain detail untuk memodelkan level game individu (non multiplayer). Pada *game flow*, dapat diketahui aliran *game play* antar scene dan dalam scene pada level game yang diberikan. Pembuatan *game flow* dilakukan pada diagram use case. Jadi diagram *game flow* bisa dianggap sebagai ekstensi dari diagram use case

Diagram *Game flow* menunjukkan proses interaksi antara player dan objek-objek yang terlihat di diagram use case (pada penjelasan skenario use case). Interaksi tersebut diwujudkan dalam bentuk pseudocode. Contohnya pernyataan IF..THEN. ELSE.

Berikut adalah contoh dari Game Flow Diagram. Gambar di bawah adalah contoh diagram compute game-flow-design untuk scene 1 on track, level "medal of honour": frontline computer game.

THE FIRST CORRUPTED VINE DIES

BLACK SMOKE begins to rise from the corpse. The smoke touches and withers adjacent shrubs and grass. [Player] looks at her weapon. A wisp of black smoke rises from the weapon.

ANY CORRUPTED VINE DIES
(INTERACTIVE)

A small cloud of black smoke rises from the vine's corpse. After a few seconds, the cloud fades away. If a player touches the black cloud, she gains a DEMON BLOOD item. Using this item causes black smoke to rise from her weapon. On the player's next hit, the black cloud envelopes the opponent, which drains life, and then fades away.

Figure 12. Penggalan Contoh *Scriptwriting*(2)

D. *Storyboard*

Storyboard adalah sketsa bagaimana mengatur cerita dan isinya [10]. *Storyboard* membantu pengembang untuk mendefinisikan parameter cerita dengan resource dan waktu yang tersedia, mengorganisasi cerita secara fokus, dan menentukan media apa yang akan digunakan pada bagian setiap cerita.

Ext. Corrupted Tree of Life - Day (Intro)

FADE IN:

Underneath an imposing BLACK CLOUD is an even more imposing sight: The CORRUPTED TREE OF LIFE. The great oak's limbs span about a hundred feet in diameter. The trunk itself is thirty feet in diameter

Figure 11. Penggalan Contoh *Scriptwriting*(1)

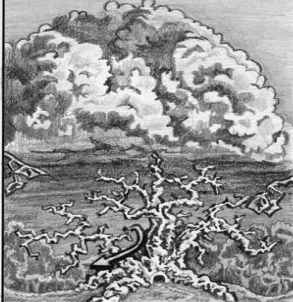
GAME TREE OF LIFE		SEQUENCE 2	SCENE 1	PAGE 1 / 12
		BOARD ID TOL-0002-001	ARTIST D. KENNERLY	DATE 2/16/04
SHOT	IMAGE	DESCRIPTION / INTERACTION		TIME
1		ESTABLISHING SHOT on BLACK CLOUD. FX: Lightning flicker inside cloud. SFX: Thunder subsides.		6
		CRANE DOWN from black cloud to TREE. Silhouette of LARGE BIRD glides down from a limb. SFX: Distant hawk screech.		4

Figure 13. Penggalan contoh *storyboard*

3 Analisa dan usulan model pengembangan game

A. Pertimbangan pemilihan format

1. Pengembangan game memerlukan lebih banyak penjelasan objek multimedia seperti gambar dan suara.
2. Pengembangan game lebih mementingkan identifikasi objek yang meliputi nama kelas, atribut, dan method.
3. Diagram *robustness* sangat membantu identifikasi objek
4. Setelah diagram sekuens dan diagram *robustness* dibuat maka diagram kelas dapat dibuat. Input dari diagram kelas adalah model domain (kalau ada), daftar objek dan *controller* pada diagram *robustness* dan daftar message pada diagram sekuens. Model domain dan daftar objek akan membentuk nama dan atribut kelas sedangkan daftar message akan menghasilkan method kelas pada diagram..

B. Usulan format pengembangan game

Dari 4 pertimbangan tersebut maka dibuatlah usulan format pengembangan game yang baru. Dokumen pengembangan game meliputi:

1. Spesifikasi kebutuhan game
2. Diagram use case dengan tambahan Diagram Desain *Game Flow*
3. Diagram robustness
4. Diagram Sekuens
5. Diagram aktivitas
6. Diagram kelas
7. *Scriptwriting*
8. *Storyboard*

Untuk mengidentifikasi kelas (membuat diagram kelas), cukup diperlukan diagram use case, diagram *robustness* dan diagram sekuens. Model domain secara eksplisit tidak perlu dibuat karena secara implisit telah ditemukan juga pada proses pembuatan diagram robustness. Proses yang terjadi pada game lebih mudah dilihat pada *storyboard* dibandingkan pada diagram aktivitas, dan diagram transisi state. Dari penjelasan di atas, perlu disyaratkan bahwa ada format desain *game flow*. Desain *Game flow* sendiri merupakan pengembangan dari diagram use case. Diagram

aktivitas (laporan indikator di referensi [3]) , merupakan 'pasangan' diagram dari diagram sekuens. Hal ini terbukti dari jumlah persentase penggunaan diagram aktivitas dan diagram sekuens sama-sama 54%. Sedangkan diagram transisi state tidak perlu dimasukkan karena jumlah penggunaan hanya 9%. *Storyboard dan scriptwriting* untuk game dibuat untuk membangun jalan cerita game dan memperkuat aspek konten multimedia game.

4 Kesimpulan

Format pengembangan game di Politeknik Batam berisi diagram UML yang meliputi diagram use case, diagram *robustness*, diagram sekuens, diagram aktivitas dan diagram kelas. Pada diagram use case ditambahkan diagram *game flow* untuk mengakomodasi aspek-aspek multimedia. Dokumen lain yang perlu dibuat adalah *scriptwriting* dan *storyboard* game. Format baru tersebut bisa mengakomodasi dua aspek dari game yaitu game sebagai software dan game sebagai produk multimedia sehingga mempermudah mahasiswa untuk mengembangkan game secara berkualitas

Pustaka

- [1] MCCONNELL, S. 2001. Who needs software engineering. IEEE Software 18 , 1 (2001), 5-8
- [2] BETHKE, E. 2003. Game Development and Production. Wordware Publishing, Plano, TX.
- [3] Roger S. Pressman. *Software Engineering: A Practioner's Approach* (Sixth Edition, International Edition). McGraw-Hill, 2005.
- [4] M. J. TAYLOR, D. GREY, and M. BASKETT, Computer Game-Flow Design, School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool, U.K, <http://www.itu.dk/courses/MSU/F2009/p3a-taylor.pdf>, diakses 23 Juli 2013
- [5] Petre, Marian (2013). UML in practice. In: 35th International Conference on Software Engineering (ICSE 2013), 18-26 May 2013, San Francisco, CA, USA (forthcoming), pp. 722-731.

- [6] Silabus Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika, Politeknik Negeri Batam, 2011
- [7] Doug Rosenberg, Use Case Driven Object Modeling, <http://www.iconixsw.com>, diakses 28 desember 2013
- [8] “How to write a screen play”,. <http://www.screenwriting.info/01.php>, di akses 1 maret 1014
- [9] Ethan kennerly, Game Script and Storyboard Creation,<http://finegamedesign.com/script/>. Diakses 1 Maret 2014
- [10]Jane Stevens, multimedia storytelling, <http://multimedia.journalism.berkeley.edu/tutorials/starttofinish/storyboarding/>, diakses 1 maret 2014