

BAB III

ANALISIS DAN PERANCANGAN

III.1. Analisa

Perkemangan *game* dari skala kecil maupun besar sangat bervariasi yang dapat dimainkan oleh siapa saja tanpa memandang umur, dari anak-anak hingga orang dewasa. *Game* berkembang begitu pesat dengan jenis *platform* yang beragam mulai dari *console*, *mobile*, *PC* dan lain sebagainya, serta dapat dimainkan secara *online* maupun *offline*.

Salah satu permainan komputer yang cukup menarik adalah permainan *Flying Pegasus* dimana dalam permainan ini pemain berusaha untuk melewati tiang penghalang untuk mendapatkan skor tertinggi di setiap levelnya. Saat ini perkembangan *game 2* dimensi untuk *platform PC* sudah sangat berkurang akibat perkembangan pesat yang terjadi pada *mobile game* khususnya *smartphone android*. Perancangan aplikasi *game Flying Pegasus* ini bertujuan untuk menumbuhkan kembali minat pengguna untuk memainkan *game 2* dimensi dengan *platform PC*. *Game Flying Pegasus* ini dibangun menggunakan bahasa pemrograman *Action Script*, yang merupakan bahasa pemrograman yang dipakai dalam *Adobe Flash*.

III.2. Analisis Kebutuhan Sistem

Perancangan bertujuan untuk memberikan gambaran logika sistem yang baru serta garis besar kepada pemakai sistem dalam desain sistem tersebut sudah tergambar logika dari seluruh sistem. Implementasi perancangan program dan pembuatan game

”*Flying Pegasus*” merupakan tahap paling penting dimana sistem yang sudah dirancang, diimplementasikan untuk menghasilkan sistem yang sesuai dengan yang diinginkan dan siap dioperasikan pada keadaan yang sebenarnya.

Pembuatan game ini membutuhkan serangkaian peralatan yang dapat mendukung kelancaran proses perancangan dan pembuatan game. Berikut ini aspek-aspek yang di butuhkan dalam pembuatan game :

III.2.1. Perangkat Keras (*Hardware*)

Hardware merupakan komponen yang terlihat secara fisik, yang saling bekerjasama dalam pengolahan data. Perangkat keras (*hardware*) yang digunakan meliputi :

- a. Monitor 14 “ LCD
- b. Memori 2 GB
- c. VGA Card Nvidia GT520 1 GB
- d. Keyboard dan mouse

III.2.2. Perangkat Lunak (*Software*)

Software adalah instruksi atau program-program komputer yang dapat digunakan oleh komputer dengan memberikan fungsi serta penampilan yang diinginkan. Dalam hal ini, perangkat lunak yang digunakan penulis untuk aplikasi perancangan dan pembuatan game adalah:

- a. Sistem Operasi Windows 7
- b. Adobe Flash CS 6

III.3. Desain Sistem

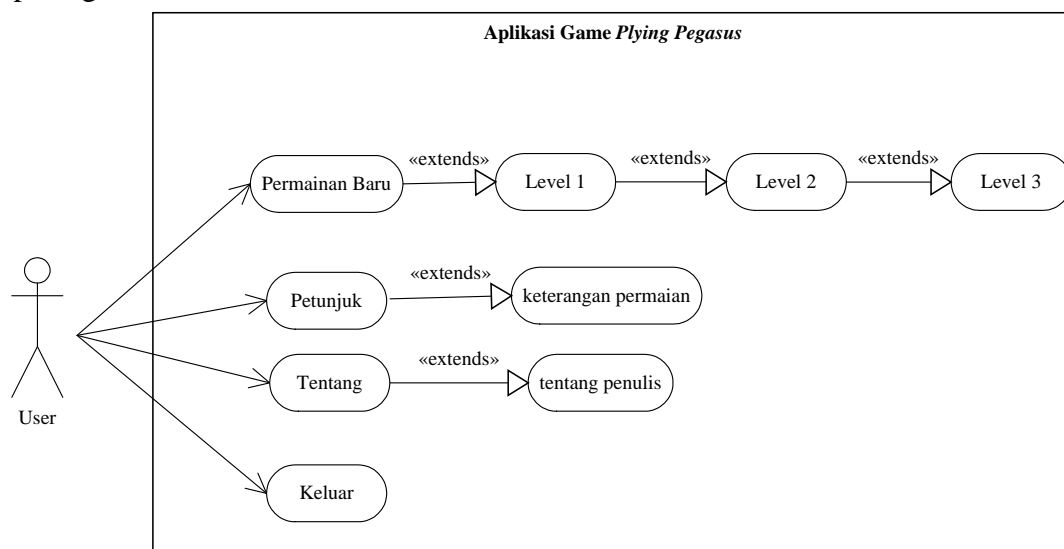
III.3.1. Use Case

Use Case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem dapat paham dan mengerti mengenai kegunaan sistem yang akan dibangun. *Use Case* digram adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut, sehingga pembuatan *Use Case* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan pada alur atau urutan kejadian.

Cara menentukan *Use Case* dalam suatu sistem :

- Pola perilaku perangkat lunak aplikasi
- Gambaran tugas dari sebuah *actor*
- Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada *actor*.
- Apa yang dikerjakan oleh suatu perangkat lunak? (“bukan bagaimana cara mengerjakannya”).

Use Case pada perancangan aplikasi *Game Flying Pegasus* dapat dilihat seperti pada gambar berikut



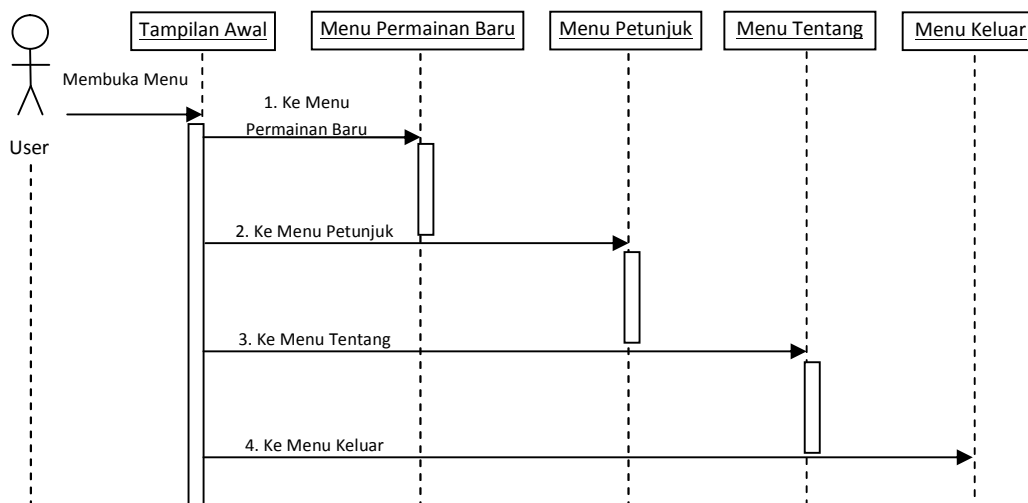
Gambar III.1 Use Case Diagram Aplikasi Game Flying Pegasus

III.3.2. Sequence Diagram

Sequence Diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, pesan/*message* dan sebagainya.

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk menghasilkan *output* tertentu. *Sequence Diagram* diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Diagram ini secara khusus berasosiasi dengan *Use Case Diagram*, *Sequence diagram* juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case*.

Sequence Diagram pada sistem aplikasi *Game Flying Pegasus* dapat dilihat seperti gambar di bawah ini :

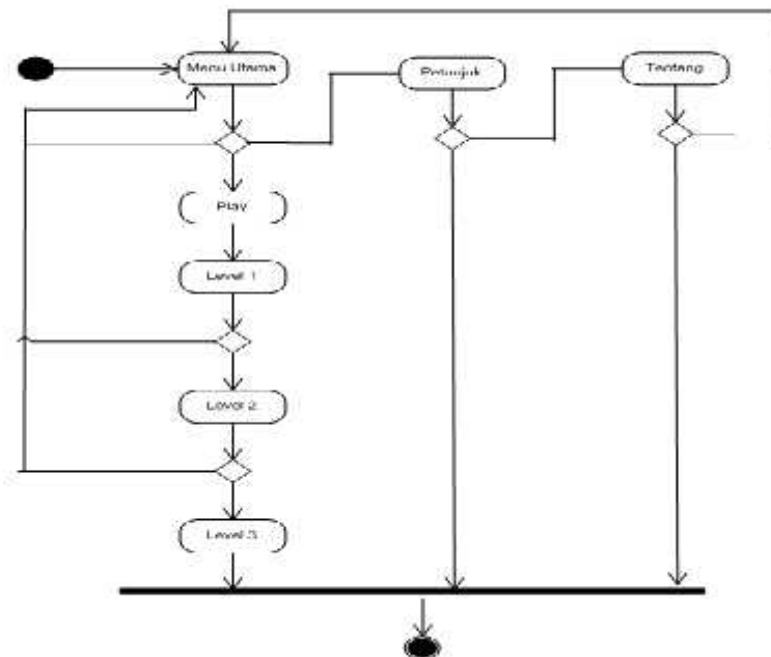


Gambar III.2 *Sequence Diagram* Aplikasi *Game Flying Pegasus*

III.3.3. Activity Diagram

Activity Diagram menggambarkan berbagai alur aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir, *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi dipicu oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *Activity Diagram* tidak menggambarkan sifat internal sebuah sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level secara umum. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case diagram*.

Activity Diagram pada sistem aplikasi game *Flying Pegasus* dapat dilihat seperti gambar dibawah ini:



Gambar III.3 Activity Diagram Aplikasi Game Flying Pegasus

III.4. Metode LCM (*Linear Congruent Method*)

LCM (*Linear Congruent Method*) merupakan metode pembangkit bilangan acak yang membangkitkan bilangan acak semu, karena pembangkitannya menggunakan operasi – operasi aritmatika. Banyak algoritma atau metode yang di gunakan untuk membangkitkan bilangan acak .*Linear Congruent Method* (LCM) merupakan metode pembangkitkan bilangan acak yang banyak digunakan dalam program komputer.LCM memanfaatkan model *linear* untuk membangkitkan bilangan acak yang didefinisikan dengan :

$$I(n+1) = (aI(n) + c) \text{ mod } m$$

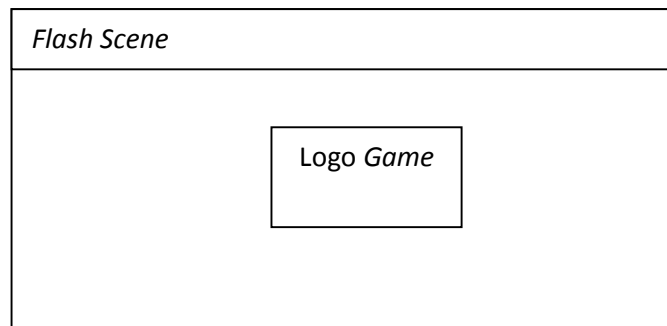
Dimana: n = adalah bil acak ke n , a dan c adalah konstanta Linear Congruent Method. m adalah batas maksimum bilangan acak. Keunggulan dari algoritma ini adalah kecepatannya yang baik, dikarenakan operasi yang dilakukan hanyalah beberapa operasi manipulasi bit saja.

III.5. Perancangan *Interface/Tampilan*

Perancangan *interface* atau tampilan pada Aplikasi *Game Flying Pegasus* ini meliputi perancangan *flash scene* atau layar pembuka, menu utama, permainan baru, petunjuk, dan tentang.

1. Layar *Flash Scene*

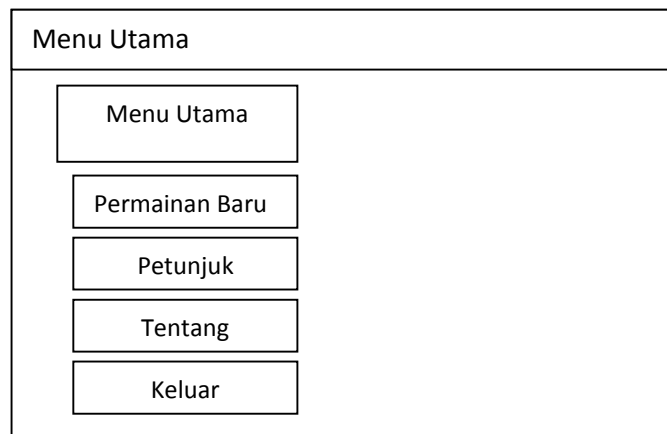
Layar ini merupakan layar pembuka pada Aplikasi *Game Flying Pegasus* yang menampilkan logo pembuat *game* ini.



Gambar III.9 Rancangan Tampilan *Flash Scene*

2. Layar Menu Utama

Layar ini merupakan layar tampilan menu utama dari Aplikasi *Game Flying Pegasus*. Pada layar ini terdapat pilihan seperti permainan baru, petunjuk, tentang, dan keluar.



Gambar III.10 Rancangan Tampilan Menu Utama

3. Layar Permainan Baru

Layar ini merupakan layar dimana *game* akan dimainkan. Begitu pengguna memilih menu ini maka layar level 1 *game* ini akan ditampilkan, pada bagian atas kanan terdapat skor, pada bagian tengah atas layar terdapat nyawa dan pada bagian kanan atas

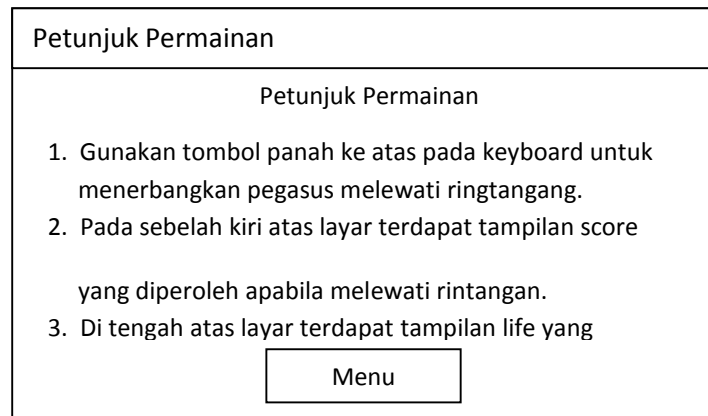
terdapat tampilan level. Untuk perhitungan skor, setiap kali *Pegasus* melewati tiang penghalang, maka skor akan bertambah satu. Dalam *game* ini terdapat tiga level permainan dimana level 1 mempunyai total skor 10, level 2 mempunyai total skor 20, level 3 mempunyai total skor 30. Dalam *game* ini juga terdapat 5 nyawa yang menentukan banyaknya kesempatan bermain yang diberikan. Apabila *Pegasus* menabrak tiang penghalang atau terjatuh melewati layar bawah *game* maka nyawa akan berkurang 1 dan permainan di ulang dari awal level yang sedang dimainkan. Dan apabila nyawa bernilai 0 maka layar *game over* akan muncul dan *game* akan di ulang dari level pertama.

Permainan Baru		
Skor	Nyawa	Level
		Objek Tiang Penghalang
Objek Flying Pegasus		Objek Tiang Penghalang

Gambar III.11 Rancangan Tampilan Permainan Baru

4. Layar Petunjuk Permainan

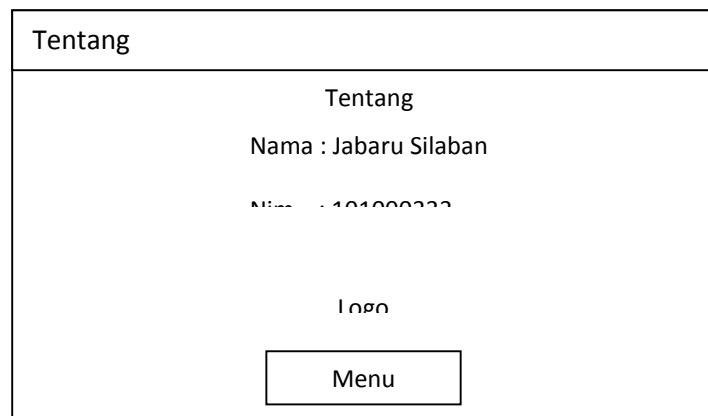
Layar ini akan menampilkan petunjuk *Game Flying Pegasus*, di layar akan dijelaskan cara memainkan *game* menggunakan tombol pada *keyboard*.



Gambar III.12 Rancangan Tampilan Petunjuk Permainan

5. Layar Tentang

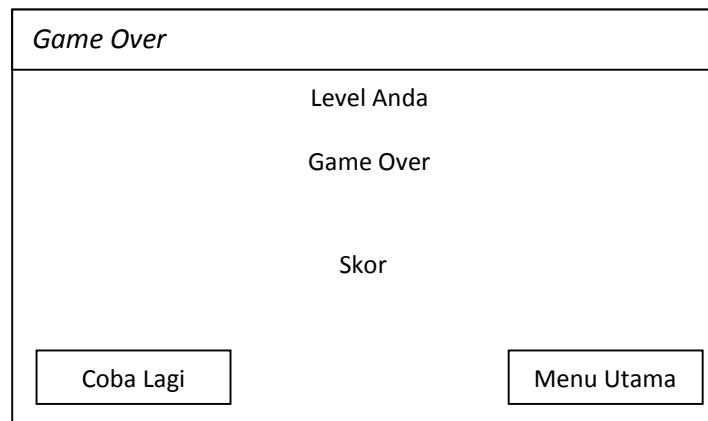
Layar ini menampilkan informasi tentang pembuat game.



Gambar III.13 Rancangan Tampilan Tentang

6. Layar *Game Over*

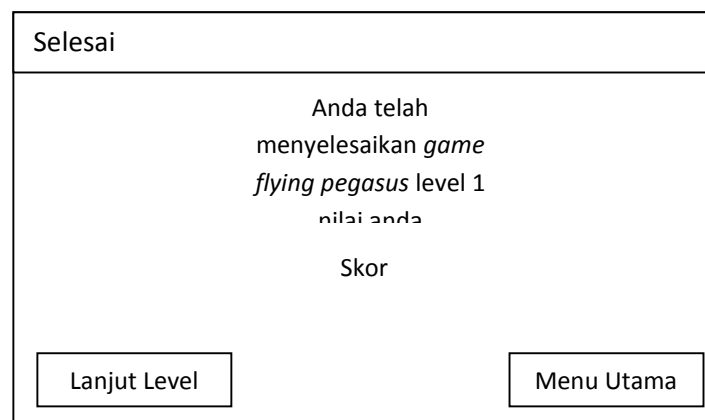
Layar ini akan tampil apabila pengguna gagal melewati tiang penghalang dalam *game*.



Gambar III.14 Rancangan Tampilan *Game Over*

7. Layar Selesai

Layar ini akan muncul apabila kita berhasil menyelesaikan sebuah level pada *game* ini



Gambar III.15 Rancangan Tampilan Selesai

III.6. Perancangan Skenario Permainan

Pada Aplikasi *Game Flying Pegasus* ini skenario permainan disusun oleh beberapa hal sebagai berikut.

- a. Seekor makhluk mitologi kuno *Pegasus* berusaha terbang kembali ke tempat para dewa Yunani kuno. Akan tetapi dia harus menghindari tiang reruntuhan bangunan kuil untuk sampai ke tempat tersebut.
- b. *Pegasus* akan terbang diantara dua tiang yang posisinya berbeda-beda. Tujuan utamanya adalah menghindar agar tidak menabrak tiang penghalang untuk mendapatkan skor.

Apabila *Pegasus* menabrak salah satu tiang tersebut maka permainan akan berakhir, dan jumlah nyawanya berkurang satu. Pengguna dapat mencoba hingga lima kali percobaan.