

# Bab 1 Pendahuluan

## 1.1 Latar Belakang

*Game* adalah sebuah permainan video yang digunakan sebagai alat hiburan sehari-hari yang bisa digunakan secara individu atau dalam sebuah kelompok kecil yang biasanya dibatasi antara 2 hingga 4 pemain. *Game* merupakan aplikasi yang tidak asing lagi bagi masyarakat baik dari anak-anak sampai dengan dewasa. Dengan perkembangan teknologi yang pesat seperti sekarang ini, *Game* menjadi semakin komplis dan juga praktis. Misalnya sekarang orang tidak perlu *Game* console untuk memainkan *Game*, *Game* dapat dimainkan di PC, laptop bahkan gadget smartphone kita bisa memainkan *Game*. *Game* juga sudah mengalami banyak perkembangan. Baik dari segi grafis, storyline yang dibangun dan lain sebagainya.

Pada awal tahun 2014, sebuah *Game* yang berukuran kecil dirilis untuk perangkat Apple IOS dan Google Android yang berjudul *Flappy bird* yang sangat sukses. *Game* itu dibuat oleh seorang *developer Game* indie Don Nguyen, *Flappy bird* berhasil mencapai Top-download chart on App Store dan juga Play Store. *Flappy bird* mengusung *Gameplay* yang sederhana, disini user akan memainkan karakter burung yang harus melewati rintangan berupa sebuah pipa, setiap kali melewati pipa user akan mendapatkan point. [1]

*Flappy bird* memberikan tantangan kepada user untuk mengendalikan karakter burung melewati terowongan pipa, yang divariasikan ketinggiannya sehingga user dapat menyelipkan melalui celah yang ada di antara pipa – pipa tersebut. Tampilan antarmuka game *flappy bird* sangat sederhana, begitu juga dengan cara memainkannya yaitu terus menekan layar untuk mengepakkan sayap, tetapi game ini sangat sulit untuk dimainkan. (theguradian, 2014).

Lalu *Game Flappy bird* akhirnya ditarik peredarannya oleh *developer*, ada beberapa alasan kenapa *developer* menarik *Game* yang sangat sukses dan menghasilkan banyak keuntungan. Salah satunya adalah user *Flappy bird* merasakan bahwa *Game* sangat sulit dan membuat frustrasi, kemudian *Game* ini juga membuat user *addictive* atau kecanduan untuk mengejar skor. Kemudian juga ada masalah hukum terkait desain *Game Flappy bird* terutama pada desain pipa yang diklaim mirip dengan desain pipa pada *Game Super Mario Bros Classic* (theguradian, 2014).

*Game Flappy Bird* dibangun dengan mekanik yang sederhana dimana pemain akan menggerakkan karakter burung untuk melawan gravitasi dan melewati rintangan yang berupa pipa-pipa yang memiliki sedikit celah. setiap berhasil melewati pipa tersebut, maka pemain akan mendapatkan *score*. Dari penjabaran mekanik *game flappy bird*, serta masalah-masalah yang timbul sampai dengan *game flappy bird* ditarik peredarannya oleh sang *developer*.

Penulis dan tim mengembangkan sebuah *game* berjudul “Gatca” adalah sebuah *game* yang dikembangkan dengan me-referensi pada *game flappy bird*, mengambil kontrol yang sama yaitu dengan tapping untuk menggerakkan karakter melawan gravitasi untuk melewati rintangan atau *obstacle*. Kemudian menambahkan *improvement* yang sebelumnya tidak ada pada *game flappy bird*, *improvement* yang dikembangkan pada *game* ini adalah menambahkan beberapa mekanik di dalam *game*, yaitu:

1. *Obstacle*
2. *Score*
3. Koin
4. *Point health*
5. 3 *Stage Area* bermain (Bumi, Mars, Neptunus)

Pengembangan *Game* ini menggunakan *Unity 3D* sebagai *Game Engine* dan menggunakan bahasa pemrograman C#, *Game* gatca dibangun untuk *platform desktop*.

Pengembangan *Game* gatca terbagi atas beberapa modul diantaranya :

1. *UI/UX Designer*
2. *Asset Designer*
3. *Developer Game*
4. *Networking Multiplayer*

Fokus utama dari laporan proyek akhir ini adalah pada bagian modul “*Developer Game*”. Sementara aspek modul lainnya pada *Game* gatca dibahas dalam dokumen terpisah.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana kontrol karakter utama dalam *Game* “Gatca”?
2. Bagaimana implementasi *Game* mekanik *Score*, koin, *Obstacle*, dan point *Health* dalam *Game*?
3. Bagaimana implementasi 3 stage area dalam *Game* “Gatca”?

## 1.3 Tujuan

Adapun tujuan dalam proyek akhir ini adalah sebagai berikut:

1. Mengimplementasikan Tapping kontrol pada karakter utama dalam *Game* “Gatca”.
2. Mengimplementasikan *Game* mekanik *Score*, Koin, *Obstacle*, dan juga Point *Health* ke dalam *Game* “Gatca”.
3. Mengimplementasikan 3 Stage area ke dalam *game* “Gatca”.

## 1.4 Ruang Lingkup Proyek Akhir

Ruang lingkup atau batasan masalah pada proyek akhir sebagai berikut:

1. *Game* 2-Dimensi Berbasis *Desktop*
2. Menggunakan *Software Unity3D* dengan bahasa *C#*
3. Basic dari *Game* ini adalah *Flappy bird*
4. Menggunakan *Package Zetcil* sebagai *Framework* pada *Game* ini
5. Hanya berfokus pada pengembangan *script Controller* di dalam *Game*, *Game Flow*, dan juga mekanik di dalam *Game*.
6. Berfokus pada pembangunan *Gameplay Single player*.
7. *Asset Designer*, *UI/UX*, dan juga *networking* dikerjakan oleh anggota kelompok lain.

## 1.5 Metodologi Pengerjaan

Proyek akhir ini dikerjakan dengan mengikuti langkah-langkah sebagai berikut:

1. *Prototype* : Titik awal pembuatan *Game* yaitu desain awal, konsep yang akan dibangun seperti apa.

2. *Pre-Production* : Kemudian berdasarkan pada hasil desain awal dan konsep pada tahapan *Prototype* lanjut membuat dokumentasi dalam bentuk dokumen desain *Game*.
3. *Production* : Tahapan pembuatan dan pengembangan asset, source code program dan juga mengintegrasikan seluruh aspek di dalam *Game*, saat sudah siap, lanjut ke tahap selanjutnya.
4. *Testing* : Pengujian terhadap fungsionalitas *Game*.
5. *Live* : *Game* sudah melewati pengujian dan sudah siap untuk dimainkan,

Langkah-langkah di atas dilakukan berdasarkan metode *Arnold Hendrick's GDLC* yang dapat alurnya dapat dilihat pada gambar 1.5.1.



Gambar 1.5.1 Alur Kerja Arnold Hendrick's GDLC

## 1.6 Waktu Pengerjaan

Waktu pengerjaan yang dipaparkan di sini adalah waktu pengerjaan PA yang sesungguhnya dilakukan dan bukan yang jadwal pengerjaan yang direncanakan dalam proposal. Dalam sub bab ini dipaparkan unit-unit pekerjaan yang dilakukan dalam berdasarkan urutan pengerjaan dan waktu pengerjaan masing-masing dalam sebuah Gantt *chart* dengan satuan waktu terkecil 1 minggu. Jika ukuran Gantt *chart* terlalu besar maka dapat diletakkan di bab Lampiran. Contoh waktu pengerjaan dapat dilihat pada lampiran A.