

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan Teknologi dan Informasi semakin hari semakin berkembang dengan pesat dalam berbagai aspek yang ada. Seiring dengan itu, perangkat komputer telah mendominasi kehidupan manusia dengan segala macam fasilitas yang ditawarkan. Salah satunya yaitu desktop komputer berteknologi *Intel®RealSense™*. *Intel®RealSense™* (dulu namanya *Intel Perceptual Computing*) adalah teknologi dimana kita dapat berinteraksi tidak hanya dengan perangkat tetapi dengan dunia di sekitar, dengan mengontrol gerakan dan pemindaian dalam dua dimensi dan tiga dimensi. Para penggemar *game* tentunya juga menginginkan perkembangan yang tidak kalah majunya dengan teknologi yang ada, sehingga kita dapat berinovasi dengan menambahkan berbagai elemen teknologi tersebut ke dalam pembuatan *game* sehingga dapat lebih menarik minat para pengguna (*user*).

Proses pembuatan *game* memerlukan sebuah *software/tool* yang disebut *game engine*. *Game engine* memberikan kemudahan dalam menciptakan konsep sebuah *game* yang akan dibuat. Unity merupakan salah satu mesin permainan (*game engine*) *cross-platform*. Dengan program Unity kita bisa membuat *game* yang dapat dimainkan pada perangkat komputer, ponsel pintar Android, *web games* (memerlukan *plugin unity web player*), iPhone, PS3, dan bahkan X-BOX. Pada proyek akhir ini akan dibuat adalah Game Puzzle The Animals Berbasis Unity Development, dimana kita dapat memainkan permainan sederhana dalam sebuah *game puzzle* berteknologi Intel®RealSense™ 3D dikembangkan oleh Unity.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas bisa didapat rumusan masalah berupa:

1. Bagaimana mengimplementasikan Intel RealSense pada *game puzzle* yang menggunakan *tools* Unity?
2. Bagaimana penerapan sensor tangan dengan menggunakan kamera Intel RealSense pada *Game Puzzle the Animals* berbasis Unity?

1.3 Tujuan

Adapun tujuan dalam pembuatan proyek akhir ini adalah sebagai berikut

1. Membuat *game puzzle* menggunakan Unity Development pada desktop komputer.
2. Eksplorasi sebuah teknologi kamera Intel RealSense dengan membuat *game puzzle*.

1.4 Batasan Masalah

Aplikasi *Game Puzzle the Animals* mempunyai batasan masalah sebagai berikut

1. Menerapkan sebuah *Game Puzzle the Animals* harus menggunakan alat sebuah kamera milik Intel RealSense tiga dimensi atau laptop yang sudah *include* kamera Intel RealSense.
2. Pengembangan perangkat hanya *support* dengan Operasi Sistem Windows 8.1 64-bit dan Windows 10 64-bit.

1.5 Definisi Operasional

Berikut adalah definisi operasional:

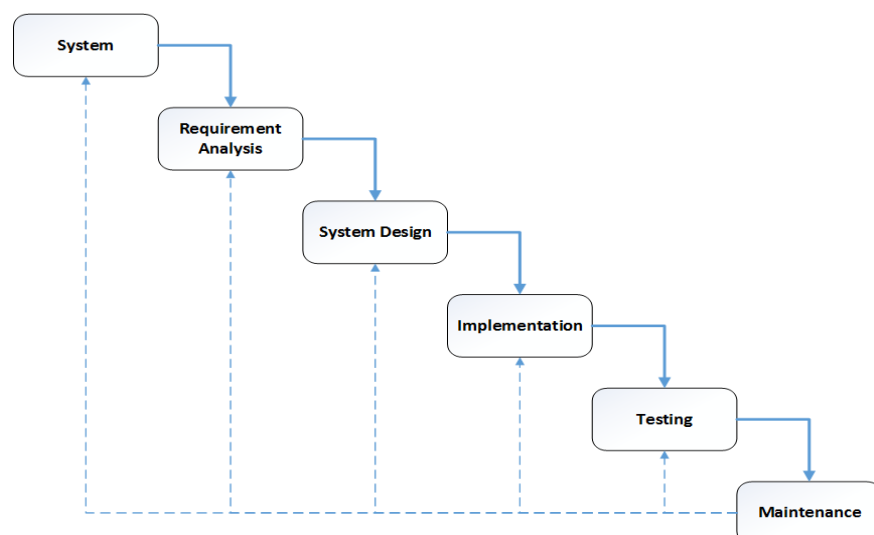
1. *Puzzle* berasal dari bahasa Inggris yang berarti teka-teki atau bongkar pasang, media *puzzle* merupakan media sederhana yang dimainkan dengan bongkar pasang. Berdasarkan pengertian tentang media *puzzle*, maka dapat disimpulkan bahwa media *puzzle* merupakan alat permainan edukatif yang

dapat merangsang kemampuan matematika anak, yang dimainkan dengan cara membongkar pasang kepingan *puzzle* berdasarkan pasangannya [1].

2. Intel®RealSense™ adalah sebuah teknologi baru yang menggunakan sebuah konsep sensor gerak tubuh manusia. Dengan menggunakan sensor tangan pada kamera Intel®RealSense™ maka permainan ini mudah digunakan.

1.6 Metode Pengerjaan

Penulis menggunakan metode *SDLC (Software Development Life Cycle)* dengan model *Waterfall*. Model *Waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut dengan “*classic life cycle*” atau model *Waterfall*. Model ini termasuk kedalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering (SE)*. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Contoh model *Waterfall* menurut Pressman, sebagai berikut [2].



Gambar 1. 1 Model *Waterfall Pressman*

Model *Waterfall* adalah suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial mulai pada tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Proses-proses tersebut dapat dijelaskan sebagai berikut:

1. *System / Information Engineering and Modeling*

Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan *hardware*.

2. *Software Requirements Analysis*

Pada tahapan ini merupakan bagian dari sistem yang terbesar dalam suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua yang diperlukan sistem, dimana nantinya program akan berjalan sesuai yang diharapkan dan menganalisis hal-hal yang diperlukan. Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*.

3. *Design*

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas sebelum *coding* dimulai. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya dengan ditambahkan *tools* yang digunakan untuk desain yaitu menggunakan *storyboard*.

4. *Coding*

Desain yang telah dibuat kemudian diubah bentuknya menjadi bahasa pemrograman melalui proses *coding*. Dalam pembuatan proyek akhir ini menggunakan Bahasa pemrograman C# (*C Sharp*) dengan *game engine* Unity.

5. *Testing/ Verification*

Semua fungsi-fungsi perangkat lunak diuji coba dengan tujuan agar perangkat lunak bebas dari *error* dan hasil perangkat lunak harus sesuai dengan kebutuhan-kebutuhan yang didefinisikan pada tahap analisis.

6. *Maintenance*

Pada tahap ini pemeliharaan terhadap software dilakukan secara berkala, apabila *software* mengalami *error* atau *bugs* yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* yang dikerjakan.

1.7 Jadwal Pengerjaan

Berikut adalah jadwal pengerjaan:

Tabel 1. 1 Tabel Jadwal pengerjaan

Keterangan	Jadwal Pengerjaan															
	Bulan Maret				Bulan April				Bulan Mei				Bulan Juni			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Sistem	■	■														
Analisis			■	■												
Desain					■	■										
Pengkodean									■	■	■	■				
Pengujian													■	■	■	
Pemeliharaan																