



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman merupakan sebuah proses pengambilan sebuah algoritma dan mengubah algoritma tersebut menjadi suatu bahasa pemrograman yang dapat dijalankan oleh komputer. Meskipun terdapat banyak bahasa pemrograman dan banyak tipe komputer, langkah terpenting pertama adalah mempunyai sebuah solusi. Tanpa algoritma program tidak ada program (Miller dan David, 2013).

Banyak pelajar Teknik Informatika yang baru pertama kali belajar pemrograman mengalami kesusahan untuk mempelajari algoritma (Jenkins, 2002). Pemrograman tidak mudah untuk dipelajari karena membutuhkan pemahaman yang benar terhadap konsep-konsep abstrak. Banyak pelajar yang kesulitan belajar pemrograman karena kekurangan bahan pembelajaran dan instruksi personal, yang mengakibatkan tingkat *drop-out* yang tinggi (Lahtinen dkk, 2005).

Salah satu cara untuk menyelesaikan masalah ini adalah dengan mengganti metode belajar. Metode yang dapat dilakukan adalah membuat sebuah proses pembelajaran yang *fun* untuk pelajar, misalnya dengan membuat proses pembelajaran tersebut menjadi suatu permainan, atau disebut juga *learning through play* (Roussou, 2004). Menurut Lahtinen dkk (2005), contoh program dilengkapi dengan visualisasi interaktif merupakan bahan pembelajaran paling berguna untuk pelajar dan pengajar. Pembelajaran algoritma dengan visualisasi melalui permainan dapat dilakukan dengan menggunakan *game* untuk mengajari algoritma (Parvini, 2011). Berdasarkan penelitian yang dilakukan oleh Kuk (2012), penggunaan *game based learning* pada pembelajaran algoritma memberikan peningkatan performa

untuk pelajar pada ujian akhir sebesar 17%, ujian praktek sebesar 26%, dan meningkatkan kepuasan dan motivasi sebesar 33%.

Quinn (2005) menyatakan bahwa suatu permainan tidak boleh deterministik secara total, diperlukan adanya *randomness* dan probabilitas agar pemain tetap tertarik dan penuh rasa ingin tahu. Menurut Walker (2014), kebanyakan angka pada *game* merupakan angka *pseudorandom*. Angka-angka ini sudah cukup acak, tapi sebenarnya tidak sepenuhnya acak. Untuk membuat angka *random*, komputer membutuhkan sebuah input, seperti input *keyboard* pemain. Akibatnya, ada kemungkinan pemain dapat memanipulasi angka-angka ini dengan melakukan tindakan spesifik. Menurut Skeet (2013), *class Random()* pada C# sendiri merupakan generator angka *pseudo-random*. *Random()* bersifat deterministik. Jika suatu instansi *Random()* dengan kondisi awal yang sama, umumnya dilakukan menggunakan *seed*, dipanggil menggunakan urutan pemanggilan yang sama, hasil yang didapatkan akan sama.

Untuk mengatasi masalah di atas, dapat digunakan metode pengacakan Durstenfeld (1964) yaitu Durstenfeld Shuffle, untuk menghasilkan angka-angka yang tidak *bias*, dengan kata lain semua angka memiliki kemungkinan yang sama untuk keluar. Keuntungan lain menggunakan Durstenfeld Shuffle adalah pengacakan dilakukan pada *array* itu sendiri, sehingga *array* baru, dengan kebutuhan memori tambahan, tidak diperlukan (Carr, 2013).

Berdasarkan pernyataan-pernyataan di atas, akan dirancang dan dibangun suatu *game* pembelajaran algoritma *sorting* dengan menggunakan Durstenfeld Shuffle sebagai pembuat soal secara acak.

1.2 Rumusan Masalah

Berdasarkan latar belakang dijabarkan beberapa masalah sebagai berikut.

1. Bagaimana merancang dan membangun game pembelajaran algoritma *sorting* dengan menggunakan *Durstenfeld Shuffle* pada *platform* Android?
2. Berapakah nilai motivasi intrinsik pengguna yang dihasilkan dari penggunaan *game* sebagai media pembelajaran pada proses pembelajaran algoritma *sorting*?

1.3 Batasan Masalah

Batasan masalah untuk penelitian ini adalah sebagai berikut.

1. Pembuatan aplikasi dilakukan menggunakan *game engine* Unity.
2. Angka-angka pada soal untuk permainan dibuat secara *random*, kemudian akan diacak menggunakan algoritma *Durstenfeld Shuffle*.
3. Aplikasi dikontrol menggunakan *touch* pada layar *Android*.
4. Aplikasi ditujukan pada mahasiswa Fakultas Teknik dan Informatika Universitas Multimedia Nusantara yang ingin mempelajari atau sudah mendapatkan pelajaran *sorting*.
5. Algoritma *sorting* yang dipelajari adalah algoritma *bubble sort*, *selection sort*, *insertion sort*, *shell sort*, *heap sort*, *merge sort*, dan *quick sort*.

1.4 Tujuan Penelitian

Tujuan yang dicapai oleh penelitian ini adalah sebagai berikut.

1. Merancang dan membangun *game* pembelajaran algoritma *sorting* dengan menggunakan *Durstenfeld Shuffle* pada *platform* Android.
2. Menerapkan penggunaan *Hedonic Motivation System Adoption Model* untuk meneliti nilai motivasi intrinsik pengguna terhadap *game* yang telah

dirancang dan dibangun dengan mengevaluasi *behavioral intention to use* dan *immersion*.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut.

1. Untuk pihak yang ingin mempelajari algoritma *sorting*, sebagai media pembelajaran algoritma *sorting* menggunakan *game*.
2. Untuk mahasiswa yang memilih jurusan *game development*, *game* yang dirancang dan dibangun bisa digunakan sebagai acuan pengaplikasian *game* pembelajaran pada *platform* Android.

1.6 Sistematika Penulisan

Sistematika penulisan laporan skripsi ini dijelaskan sebagai berikut.

1. BAB I Pendahuluan

Berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. BAB II Tinjauan Pustaka

Berisi landasan teori terkait *sorting*, Durstenfeld Shuffle, Unity, struktur *game*, *game-based learning*, motivasi pembelajaran, metode penelitian kualitatif, *purposive sampling*, *hedonic motivation system adoption model*, *likert scale*, dan Cronbach Alpha.

3. BAB III Metodologi Penelitian dan Perancangan Sistem

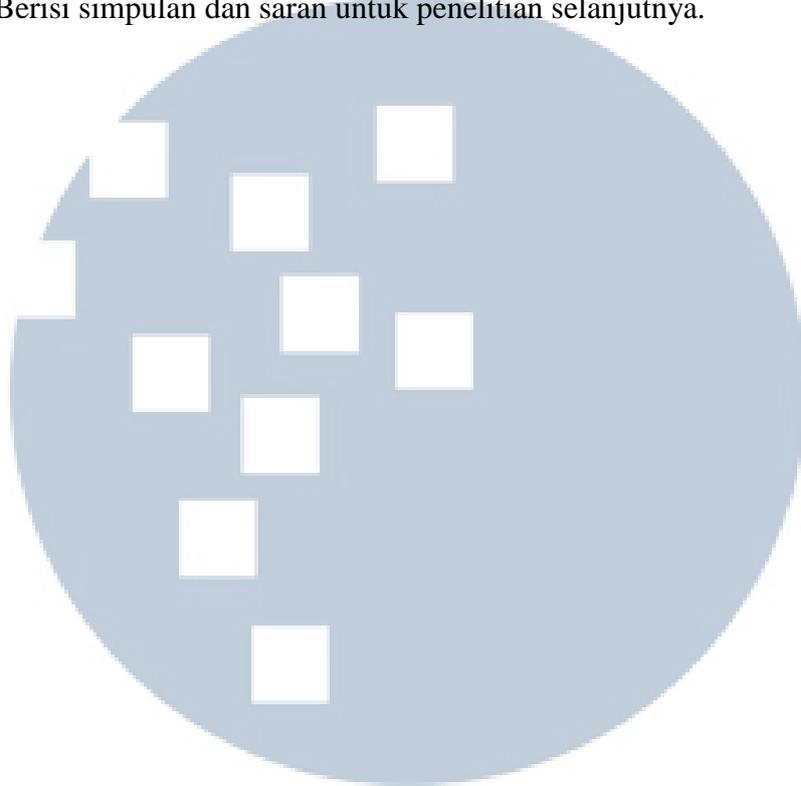
Berisi metodologi penelitian yang digunakan serta proses perancangan terkait dengan kebutuhan *game* dan desain keseluruhan *game*.

4. BAB IV Implementasi dan Uji Coba

Berisi penjelasan mengenai implementasi dan hasil uji coba *game* yang dirancang dan dibangun.

5. BAB V Simpulan Dan Saran

Berisi simpulan dan saran untuk penelitian selanjutnya.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA