

# BAB I PENDAHULUAN

## I.1 Latar Belakang

Version Control Systems atau disingkat dengan VCS telah digunakan oleh banyak *software engineer* selama pengembangan *software* karena membantu mereka mengelola *source code* dan memungkinkan mereka untuk menyimpan setiap versi proyek yang telah dikerjakan [1]. VCS sangat berguna karena mendukung kerangka kerja kolaboratif yang memudahkan *software engineer* untuk bekerja sama secara efektif. Tanpa VCS, kolaborasi akan sangat begitu sulit [2]. Oleh karena itu, penting bagi semua *software engineer* untuk memahami bagaimana cara mengoperasikan *version control system* [2]. Salah satu yang paling umum digunakan oleh *software engineer* adalah Git [3]. Pada beberapa kasus seperti membangun proyek bersama dalam sebuah tim dan menginterview kandidat *software engineer*, ditemukan *software engineer* yang masih banyak belum memahami teknik pengoperasian tertentu di *Git* dari tingkat yang dasar sampai lanjut [4].

Berdasarkan wawancara yang dilakukan pada salah satu *startup* di Bandung pada 18 Oktober 2022 lebih dari 60% dari total kandidat sejumlah 78 *software engineer* yang melamar kepada perusahaan tersebut gagal disebabkan karena kurangnya pengetahuan terkait Git baik dari teori maupun praktik (Lampiran 1). Selain itu, dalam fase pengembangan di *startup* tersebut beberapa *software engineer* pada level junior ditemukan pernah mengalami masalah terkait Git yang tidak bisa di tangani sendiri, yaitu mengenai *command line* dasar yang seharusnya bagi *software engineer* di level mana pun harus dapat melakukannya. Maka saat masalah tersebut ditemukan *software engineer* dari tim lain perlu membantu *software engineer* tersebut yang mengalami kesulitan yang mengakibatkan kemajuan *development* menjadi kurang efisien karena *software engineer* pada tim lain ikut sibuk mengurus urusan di luar tanggung jawabnya (Lampiran 1). Berdasarkan data wawancara tersebut dapat disimpulkan bahwa tingkat pemahaman teori dan pengoperasian Git dari *software engineer* masih rendah. Untuk meningkatkan pemahaman VCS Git, *startup* tersebut melakukan *peer mentoring* dari *engineer* yang cukup pengalaman dengan yang belum. Namun, metode *peer mentoring* mengalami kendala di mana

pemmentor akan terbagi waktunya untuk melatih *software engineer junior* dan menyita waktu pemmentor untuk melanjutkan pekerjaan lainnya dan juga sesi *mentoring* tersebut belum memberikan dampak yang cukup signifikan terhadap pemahaman *software engineer junior*.

Berdasarkan permasalahan tersebut, diperlukan adanya alat bantu pembelajaran yang dapat membantu *software engineer junior* berlatih meningkatkan pemahamannya secara mandiri. Penelitian ini bertujuan untuk membangun alat bantu pembelajaran VCS Git berbasis *serious game*. Pendekatan *serious game* dipilih karena berdasarkan dari beberapa penelitian *serious game* terbukti dapat meningkatkan pemahaman, ketangkasan, dan motivasi maupun *engagement* dari metode pembelajaran konvensional [5]–[7].

Untuk mengetahui aplikasi *serious game* yang baik, dilakukan audit kompetitor pada aplikasi pembelajaran (Lampiran 2). 3 data *serious game* yang diaudit merupakan *serious game* dengan materi ilmu komputer dengan ketentuan 5 tahun terakhir dari tanggal waktu rilis dan dibandingkan kriteria-kriterianya berdasarkan penelitian Caserman tentang kriteria kualitas untuk *serious game* [8], didapatkanlah hasil terbaik yaitu 7 Billion Humans yang akan digunakan sebagai acuan pembuatan *serious game* berbasis pengetahuan *verison control system*. Agar acuan pembuatan *serious game* lebih terukur, maka pada *game* tersebut perlu ditinjau lebih lanjut mengenai apa saja pola perancangan yang diterapkan, *game design pattern* membantu dalam permasalahan tersebut yang di mana mempunyai arti lain sebagai deskripsi interaksi berulang yang relevan dengan *gameplay* [9]. *Game design pattern* disimpulkan juga pada penelitian Björk sebagai suatu komponen yang membantu proses pengembangan sebuah *game* [9], [10].

Metode yang akan digunakan untuk pengembangan aplikasi pembelajaran VCS Git ini adalah metode *waterfall*, karena metode ini mempunyai karakteristik proses pengembangan yang diselesaikan setiap fasenya satu per satu mulai dari mendefinisikan *requirement*, memodelkan desain, implementasi, pengujian fungsionalitas aplikasi dan *maintenance* (jika ada) [11]. Karakteristik proses pengembangan seperti itu cocok terhadap aplikasi yang akan dibuat karena pada penelitian ini mempunyai keterbatasan akses untuk terus berkomunikasi terhadap

subjek penelitian dan keterbatasan waktu untuk mengevaluasi *requirement* setiap fase tertentu selama proses pengembangan.

Aplikasi *serious game* pembelajaran VCS Git yang telah dibangun, akan dievaluasi dampaknya terhadap tingkat pemahaman menggunakan metode kuasi-eksperimen. Alat ukur yang digunakan untuk mengukur aspek tersebut adalah tes sumatif yang akan dirancang untuk memuat pertanyaan dengan konteks pengetahuan VCS Git. Tes tersebut akan diujikan terhadap 2 kelompok yang berbeda, kelompok kontrol akan belajar melalui sumber yang konvensional seperti modul atau video dan kelompok eksperimen belajar melalui aplikasi pembelajaran VCS Git. Tes sumatif akan diberikan sebelum dan sesudah mempelajari VCS Git pada masing-masing grup sehingga terlihat selisih skor sebelum dan sesudah diujikan.

## **I.2 Rumusan Masalah**

1. Bagaimana menerapkan fitur *serious game* berdasarkan *game design pattern* pada aplikasi pembelajaran VCS Git.
2. Bagaimana dampak aplikasi *serious game* VCS Git terhadap tingkat pemahaman pada *fresh graduate software engineers*?

## **I.3 Tujuan**

1. Merancang *game design* berdasarkan *game design pattern* yang didefinisikan pada batasan penelitian kedua untuk diterapkan pada aplikasi.
2. Membuat aplikasi *serious game* yang memuat pembelajaran VCS Git menggunakan metode pengembangan *waterfall* yang nantinya akan diuji tingkat pemahamannya pada *fresh graduate software engineers* menggunakan tes sumatif dengan metode kuasi eksperimen.

## **I.4 Batasan Penelitian**

1. Ruang lingkup pembelajaran VCS Git yang akan diimplementasikan adalah *git command* yang hanya berkaitan dengan repositori lokal (tidak termasuk repositori *remote*) sebagai berikut: *commit, cherry-pick, branch, reset, revert, merge, rebase, checkout, tag, dan describe*.

2. Batasan *game design pattern* yang akan dijadikan acuan adalah *block-based programming, step by step program execution, visual Input and output representation, hidden game programming challenges, program examples levels, program output log*.

Dasar pemilihan *git command* berdasarkan *command-command* yang sering dipakai pada industri pengembangan perangkat lunak.

Sedangkan dasar pemilihan *game design pattern* didasari hasil audit kompetitor pada lampiran 2 dan analisis pada *comparative study* di Bab II yang menunjukkan bahwa *game design pattern* tersebut paling tepat untuk pengembangan aplikasi *serious game* pembelajaran VCS Git.