

1. Pendahuluan

Latar Belakang

Implementasi metode *Scrum* memiliki banyak keuntungan dalam tim[1]. Beberapa diantaranya adalah meningkatkan komunikasi antar anggota tim, meningkatkan kualitas produk, dan mudah beradaptasi terhadap perubahan *requirement* seiring proses *software development*[2]. Namun, terdapat tantangan dan kekurangan yang akan dihadapi dalam implementasi metode *Scrum*[2][3]. Salah satu permasalahan ada dalam *Code Quality*[3]. Dalam proses *Scrum*, tim hanya memiliki waktu yang singkat dalam setiap *sprint* karena *Scrum* menuntut kecepatan dalam proses disetiap *sprint*[4]. Hal ini menyebabkan tim tidak bisa melakukan *code review* secara menyeluruh, mengakibatkan tim harus menambah beban dan tekanan kerja mereka[3]. Oleh sebab itu, *Code Quality* dari produk yang dikembangkan menurun dan menghasilkan masalah yang kritis, seperti program tidak mudah untuk di *maintain*[3].

Code review diperlukan untuk mengidentifikasi kecacatan dalam kode yang ditulis sebelum masuk kedalam inti proyek[5]. Tetapi sifat rumit, memakan waktu dan tenaga dalam *code review* menyebabkan penghambatan dalam mengadopsi praktik *code review* dengan benar[5][6]. Untuk mempermudah dan mempercepat *code review*, Carmine, et, al.,[7] menjabarkan metode *Continuous Code Quality*(CCQ) untuk menangani permasalahan terkait *code quality* secara general dan mempercepat proses *code review*.

Penerapan *Scrum* menurut *Scrum guide*[8] tidak menunjukkan adanya praktik CCQ. Didasari oleh hal tersebut penelitian ini akan berfokus untuk menambahkan praktik CCQ pada proses *Scrum* dan melihat efek implementasi CCQ pada *Scrum* serta *velocity* pada setiap *sprint* seiring proses *software development*. Penelitian akan dilakukan secara empiris melalui eksperimen case study dalam *software development*. Selanjutnya, akan dilakukan analisa terhadap penambahan praktik CCQ pada metodologi *Scrum* dengan melakukan *interview* dengan tim *Scrum* untuk mengetahui apakah CCQ memberikan efek positif atau justru tidak memecahkan permasalahan yang diangkat. Serta akan dilakukan perhitungan *velocity* dengan melihat jumlah *story point* yang diambil pada setiap *sprint* untuk mengetahui kecepatan proses pada *scrum*.

Pada penelitian ini tidak akan berfokus untuk meneliti fitur, Bahasa pemrograman dan tools yang digunakan, dengan tujuan untuk lebih fokus dalam penelitian proses dan eksperimen yang akan dilakukan dengan tim pengembang.

Topik dan Batasannya

penelitian ini akan berfokus dalam mengetahui efek menghubungkan praktik *Continuous Code Quality* dengan *Scrum* terhadap kecepatan pada masing-masing *sprint* serta mengetahui pengaruh menggabungkan praktik *Continuous Code Quality* dengan *Scrum* terhadap kinerja tim pengembang dalam studi kasus Sistem Informasi Klinik. Berdasarkan latar belakang yang diangkat, berikut rumusan masalah yang akan dibangun:

1. Apa efek menghubungkan praktik *Continuous Code Quality* dengan *Scrum* terhadap kecepatan pada masing-masing *sprint*?
2. Bagaimana pengaruh menggabungkan praktik *Continuous Code Quality* dengan *Scrum* terhadap kinerja tim pengembang?

Tujuan

Berdasarkan rumusan masalah diatas, maka didapatkan tujuan berikut:

1. Mengetahui efek menghubungkan praktik *Continuous Code Quality* dengan *Scrum* terhadap kecepatan pada masing-masing *sprint*.
2. Mengetahui pengaruh dari menggabungkan praktik *Continuous Code Quality* dengan *Scrum* terhadap kinerja tim pengembang.