

# 1 Pendahuluan

## 1.1 Latar belakang

Tingkat efisiensi pembangunan perangkat lunak mulai menunjukkan penurunan yang sangat signifikan. Begitu pula yang dialami CV 'X', sebagai perusahaan pembangun perangkat lunak skala kecil-menengah [8] mengalami peningkatan biaya operasional yang meningkat secara bertahap terlebih untuk perusahaan non berbadan hukum faktor solvabilitas sangat sensitif. Beberapa analisis internal perusahaan dan penulis sendiri berasumsi bahwa salah satu faktor pemicu permasalahan ialah metode pembangunan perangkat lunak *Waterfall* yaitu metodologi pembangunan perangkat lunak yang terurut dan sistematis, yang digunakan perusahaan sudah tidak cocok digunakan untuk situasi bisnis sekarang ini. Terlebih terdapat fakta materil yang mendukung asumsi tersebut.

Trend metode *Agile* yaitu metode pembangunan perangkat lunak yang ringan dengan mengedepankan program daripada dokumentasi, yang muncul satu dasawarsa lalu terbukti mampu membantu beberapa perusahaan meningkatkan efisiensinya. Salah satu turunan metode *Agile* yang banyak dipakai ialah *Scrum*. *Scrum* adalah metodologi pembangunan perangkat lunak agile berbasis sinergi antara kerja tim dan kebutuhan bisnis secara *iteratif* dan *incremental*. *Scrum* dalam praktik berperilaku seperti payung yang menaungi metode *agile* lain [1], sehingga beberapa *best practice* yaitu praktik-praktik di lapangan metode *agile* yang diperlukan dapat dipakai dalam lingkup *Scrum*.

Implementasi *Scrum* pada tugas akhir ini penulis mengambil beberapa *best practice Rational Unified Process (RUP)* yaitu metodologi *agile* yang menitikberatkan pada arsitektur perangkat lunak dan *Extreme programming (XP)* yaitu kumpulan *best practise agile* untuk mengatasi permasalahan dokumentasi desain dan tata cara pemrograman. Untuk membedakan *Scrum* yang terdapat dalam tugas akhir ini dengan *Scrum* versi lainnya maka *Scrum* untuk tugas akhir ini penulis namakan *F-Scrum* yaitu varian *Scrum* yang dilengkapi dengan dokumentasi perangkat lunak dan *best practices*. Untuk melihat tingkat efisiensi *F-Scrum* terhadap *Waterfall* maka digunakan pemodelan *COCOMO* yaitu model pengukuran ongkos pembuatan perangkat lunak yang terdiri dari *Relative Effort* yaitu parameter yang digunakan untuk mengukur ongkos pembuatan perangkat lunak dan *Time to Develop* yaitu parameter untuk mengukur perkiraan waktu yang diperlukan untuk membangun suatu perangkat lunak

## 1.2 Perumusan masalah

Berdasarkan uraian di atas, maka perumusan masalah yang dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana implementasi tata cara tahapan membangun perangkat lunak berbasis *Scrum*.
2. Bagaimana hasil akhir perbandingan pembuatan perangkat lunak berbasis metoda *Waterfall* dan *F-Scrum* berdasarkan parameter *Relative Effort* dan *Time to Develop*.

Untuk menjaga agar dalam penelitian tetap efektif, permasalahan tidak meluas, dan pembahasan tidak menyimpang dari tujuan semula serta menjadi mudah dipahami sesuai dengan tujuan penelitian yang hendak dilakukan, maka perlu dilakukannya pembatasan masalah sebagai berikut :

1. Spesifikasi kebutuhan perangkat lunak yang baru sama dengan perangkat lunak Aplikasi *Billing* yang telah ada sebelumnya, yang menitikberatkan pada fungsionalitas Registrasi Pelanggan Baru, *Maintenance* Data Pelanggan, Membayar Tagihan dan *Maintenance* Sistem,
2. Perangkat lunak yang dibuat memiliki karakteristik yang sama dengan aplikasi yang telah dibuat yaitu *standalone* yaitu untuk satu pengguna. Hanya konsep pembangunan perangkat lunak yang baru berorientasi objek sedangkan aplikasi terdahulu berbasis struktural,
3. Parameter perbandingan utama yang berlaku untuk struktural dan berorientasi objek ialah *Relative Effort* dan *Time to Develop*,
4. Asumsi perbedaan karakteristik pribadi dan kemampuan pembangun aplikasi terdahulu (*Waterfall*) dengan pembangun aplikasi berbasis *Scrum* diabaikan.

Berdasarkan kerangka pemikiran yang telah dikemukakan pada penjelasan diatas maka dapat ditarik hipotesis penelitian :

*Efisiensi pengerjaan aplikasi berbasis metode Scrum lebih baik dibandingkan aplikasi berbasis metode Waterfall, ditinjau dari parameter Relative Effort dan Time to Develop untuk perangkat lunak yang sama pada kasus TA ini.*

### **1.3 Tujuan**

Adapun tujuan yang hendak dicapai melalui penelitian yang dilakukan ini adalah sebagai berikut :

1. Mengimplementasikan konsep *Scrum* dalam pembangunan perangkat lunak pada praktek lapangan sebenarnya,
2. Membandingkan hasil akhir pembangunan perangkat lunak berbasis konsep *Waterfall* dan *Scrum* berdasarkan parameter *Relative Effort* dan *Time to Develop*,
3. Mengevaluasi hasil perbandingan proses pengerjaan perangkat lunak berbasis *Scrum* terhadap *Waterfall*.
4. Memberikan saran perbaikan atas konsep terapan pembangunan perangkat lunak berbasis *Scrum* khususnya untuk pengembang proyek skala kecil dan menengah.

### **1.4 Metodologi penyelesaian masalah**

Sebagai realisasi untuk mencapai tujuan dan memecahkan masalah, dilakukan langkah-langkah sebagai berikut:

1. Studi literatur berhubungan dengan metode *Scrum* dan *Waterfall*,
2. Pengambilan beberapa *best practice RUP* dan *XP* dalam lingkup *Scrum*. Hasil modifikasi *Scrum* ini dinamakan *F-Scrum*,
3. Melakukan studi lapangan untuk membuat perencanaan jadwal pembangunan perangkat lunak,
4. Menyusun *User Stories* berdasarkan masukan dari *stakeholder*,
5. Menyusun *product backlog* yang merupakan hasil translasi dari *User Stories*,

6. Memilih beberapa item *product backlog* untuk dikerjakan dalam waktu satu *Sprint*,
7. Menyusun *Use Case diagram* dan deskripsi *Use Case* terpilih untuk satu *Sprint*,
8. Memulai *Sprint*,
9. Membuat diagram kelas dan diagram *sequence* pada waktu *daily stand up meeting* untuk modul perangkat lunak yang telah selesai,
10. Melakukan *Refactoring* atas desain dan kode pemrograman,
11. Melakukan *Sprint Review* setelah satu *Sprint* selesai,
12. Ulangi *Sprint* sampai semua item *product backlog* selesai dikerjakan,
13. Membuat diagram komponen setelah perangkat lunak selesai,
14. Melakukan *Sprint Retrospective* yang didalamnya berisi evaluasi pengerjaan perangkat lunak dengan *F-Scrum*. Selain itu juga dilakukan analisis perbandingan tingkat efisiensi pembangunan perangkat lunak antara *F-Scrum* dan *Waterfall* dengan melibatkan tim *Scrum* dan tim *Waterfall*.