

BAB I

PENDAHULUAN

1.1 Latar Belakang

Requirements elicitation atau elisitasi kebutuhan adalah proses mencari, mengungkap, memperoleh, dan menguraikan persyaratan untuk proses pengembangan perangkat lunak (Coulin & Zowghi, 2005). *Requirement elicitation* merupakan proses awal dari proses pengembangan perangkat lunak (Sabahat et al., 2010). Dalam praktiknya, *requirement elicitation* merupakan proses yang memiliki peranan penting dalam pengembangan perangkat lunak, karena apabila persyaratan dalam *requirement elicitation* tidak lengkap maka dapat meningkatkan biaya dan upaya dalam pengembangan perangkat lunak (Sharma & Pandey, 2014). Ekstraksi kebutuhan yang didapat dari berbagai sumber dokumen tidak terstruktur meningkatkan efisiensi pengembangan perangkat lunak (Raharjana et al., 2019). praktik penggunaan ulang memungkinkan mengurangi biaya dan upaya dalam proses *requirement elicitation* (Schots, 2014).

Requirements reuse atau penggunaan ulang kebutuhan adalah proses penggunaan kembali kebutuhan perangkat lunak yang sudah dikembangkan dan digunakan di proyek sebelumnya. *Requirement reuse* merupakan metode yang efektif untuk meningkatkan efisiensi dalam proses pengembangan perangkat lunak, mengurangi biaya pengembangan dan pemeliharaan perangkat lunak, menghasilkan produk perangkat lunak berkualitas lebih tinggi, meningkatkan ketergantungan perangkat lunak dan sistem serta mempermudah transfer anggota tim, alat, dan metode antar proyek (IEEE Standards Association, 2010).

Umumnya *requirement reuse* diekstraksi dari *software requirement specification* (SRS), namun ekstraksi SRS untuk proses *requirement reuse* tidak bisa dilakukan oleh semua orang karena tidak semua orang memiliki akses ke artefak perangkat lunak (Bakar et al., 2016). Selain didapatkan dari SRS, *requirement reuse* juga dapat diekstraksi dari ulfasan pengguna dan deskripsi aplikasi. Ulasan pengguna merupakan penilaian yang ditulis oleh pengguna. Selain penilaian, beberapa pengguna menyarankan fitur baru yang diinginkan oleh pengguna. Sedangkan, deskripsi aplikasi ditulis oleh pengembang yang berisi

informasi lengkap mengenai nama aplikasi, fitur, manfaat, dan tujuan aplikasi tersebut agar pengguna dapat dengan mudah mencari aplikasi yang dibutuhkan (Johann et al., 2017).

Pengembangan perangkat lunak dapat dilakukan dengan beberapa macam metode. Salah satunya adalah metode *agile development software* yang merupakan pengembangan perangkat lunak jangka pendek sehingga membutuhkan adaptasi cepat dari pengembang terhadap berbagai jenis perubahan. Pada metode *agile*, *user story* yang dikumpulkan pada saat proses *requirement elicitation* merupakan artefak yang paling banyak digunakan untuk mendapatkan deskripsi sederhana dari perspektif pengguna (Raharjana et al., 2019). Dengan banyaknya *user story* yang digunakan pada proses *requirement elicitation*, memungkinkan *user story* dapat digunakan ulang (*requirement reuse*) untuk proyek pengembangan perangkat lunak mendatang.

User story adalah kalimat pendek yang terstruktur berisi permintaan kebutuhan atau fungsi dari pengguna (Thamrongchote & Vatanawood, 2016). *User story* terdiri dari tiga aspek yaitu siapa/*who* yang meminta kebutuhan atau fungsi, apa atau *what* kebutuhan atau fungsi yang diminta oleh *stakeholders*, dan kenapa/*why* *stakeholders* menginginkan fungsi tersebut (Dalpiaz & Brinkkemper, 2018). *User story* banyak digunakan untuk pada saat proses *requirement elicitation* karena *user story* berperan menjelaskan kebutuhan dan keinginan *stakeholders* atau persona (Lucassen et al., 2016). *User story* dapat menjadi *boundary object* dalam perpindahan dan pembuatan pengetahuan baru (O'hEocha & Conboy, 2010). *Agile software development* adalah salah satu metode yang banyak menggunakan *user story* pada proses *requirement elicitation* (Raharjana et al., 2019). *User story* yang telah dikumpulkan umumnya tidak digunakan lagi oleh pengembang perangkat lunak, disebabkan banyak pengembang belum berpengalaman dalam memanfaatkan potensi penggunaan ulang dari artefak (Dilorenzo et al., 2020). ini membuka peluang untuk pengembang menggunakan kembali *user story* yang sudah dikumpulkan dari beberapa proyek sebelumnya untuk proyek selanjutnya.

Pada penelitian ini akan dilakukan ekstraksi *software feature* dari deskripsi aplikasi. Deskripsi aplikasi memiliki spesifikasi yang sesuai dengan penelitian ini

karena deskripsi aplikasi dapat ditulis di awal proses pengembangan perangkat lunak dan dapat diakses oleh semua orang. Berbeda dengan SRS yang tidak dapat diakses oleh semua orang dan ulasan pengguna yang hanya bisa didapatkan saat perangkat lunak telah rilis, digunakan, dan di ulas oleh pengguna (Bakar et al., 2016). Kumpulan *user story* dari beberapa proyek pengembangan perangkat lunak akan digunakan ulang dengan mencari kedekatan dari masing-masing *user story* dengan ekstraksi *software feature* yang telah di ekstraksi dari deskripsi aplikasi. *user story* yang memiliki keterkaitan akan menjadi rekomendasi dalam proses pengembangan perangkat lunak. Penggunaan ulang *user story* berdasarkan deskripsi aplikasi untuk kebutuhan elisitasi perangkat lunak ditujukan untuk mengurangi biaya dan upaya dalam pengembangan perangkat lunak. Praktik penggunaan ulang (*reuse*) memungkinkan mengurangi biaya dan upaya dalam proses *requirement elicitation* (Schots, 2014). Manfaat yang dihasilkan dari efisiensi biaya dan waktu adalah kepuasan pelanggan. Di mana tolak ukur keberhasilan *agile software development* adalah berdasarkan waktu, biaya, upaya dan kepuasan pelanggan (Tam et al., 2020).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka rumusan masalah pada penelitian ini adalah bagaimana cara melakukan penggunaan ulang *user story* berdasarkan deskripsi aplikasi untuk proses elisitasi kebutuhan perangkat lunak?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah penggunaan ulang *user story* untuk proses elisitasi perangkat lunak berdasarkan ekstraksi *software feature* dari deskripsi aplikasi awal. Penelitian ini menghasilkan *output* daftar *user story* yang memiliki potensi untuk digunakan ulang. potensi didapatkan dari nilai *similarity user story* dengan ekstraksi *software feature* dari deskripsi aplikasi. Output yang dihasilkan akan diukur menggunakan *precision and recall test*.

1.4 Manfaat Penelitian

Manfaat yang ingin dicapai pada penelitian ini adalah:

1. Membantu pengembang perangkat lunak untuk menggunakan ulang *user story (software requirements)* berdasarkan deskripsi aplikasi awal

yang telah tersedia. Sehingga bisa membantu mengidentifikasi *user story* yang terkait secara lebih cepat.

2. Membantu penulis memberikan kesempatan untuk melakukan eksplorasi penerapan teori-teori dan pendekatan yang berhubungan dengan aspek pengembangan perangkat lunak dan ekstraksi *user story*.
3. Membantu pembaca dapat digunakan sebagai tambahan informasi dan pengembangan wawasan, serta dapat dijadikan referensi penelitian selanjutnya yang berhubungan dengan pengembangan perangkat lunak, *requirement engineering*, dan ekstraksi *user story* di masa mendatang

1.5 Batasan Masalah

Penelitian ini memiliki batas pengendalian pengerjaan untuk fokus pada permasalahan yang dibahas. Batasan masalah tersebut adalah sebagai berikut:

1. Kebutuhan perangkat lunak menggunakan dataset *user story* yang sudah disiapkan.
2. *User story* ditulis dalam Bahasa Inggris. Karena dataset *user story* yang sudah dipersiapkan berbahasa inggris, dan *library* dalam Bahasa pemrograman Python yang digunakan pada penelitian ini lebih optimal pada pengelolaan teks dalam bentuk Bahasa Inggris dibandingkan Bahasa Indonesia.

BAB II

TINJAUAN PUSTAKA

Pada bagian ini akan dibahas teori yang mendasari penelitian ini. Pembahasan yang ada akan menjadi landasan dasar untuk memahami permasalahan yang ada. Penelitian mengenai penggunaan ulang user story, deskripsi aplikasi, serta metode yang digunakan pada penelitian ini akan dibahas secara mendalam pada bab ini.

2.1 *User Story*

User story adalah kalimat pendek yang terstruktur berisi permintaan kebutuhan atau fungsi dari pengguna (Thamrongchote & Vatanawood, 2016). *User story* terdiri dari tiga aspek (Dalpiaz, & Brinkkemper, 2018) yaitu siapa/*who* yang meminta kebutuhan atau fungsi, apa atau *what* kebutuhan atau fungsi yang diminta oleh *stakeholders*, dan kenapa/*why* *stakeholders* menginginkan fungsi tersebut, dalam praktiknya beberapa *user story* tidak menyertakan aspek *why*. Teknik penulisan *user story* memiliki template sederhana yaitu; “*As a*/Sebagai” diikuti peran user, “*I want* / Saya ingin” diikuti permintaan fitur atau tujuan, “*So that*/Sehingga” diikuti hasil yang diharapkan atau alasan. *user story* memegang peranan utama yaitu menjelaskan kebutuhan atau keinginan dari *stakeholders* (Lucassen et al., 2016). *User story* dapat ditulis dengan format isi tiga aspek lengkap atau hanya berisi aspek *who* dan *what* saja, karena aspek *why* bersifat opsional (Raharjana et al., 2019). Secara fisik, *user story* ditulis di suatu media yang disebut *story card*. Pada penelitian ini *user story* akan di ekstrak untuk mendapatkan ini dari *user story*, ekstraksi dilakukan dengan mengelompokkan aspek *who*, *what* dan *why* pada tiap *user story*. Contoh bentuk *user story* dapat dilihat di tabel 2.1.