

BAB 1. PENDAHULUAN

1.1 Latar Belakang Masalah

Perangkat lunak menurut Pressman adalah sebuah perintah program dalam komputer, yang apabila dieksekusi oleh penggunanya akan memberikan fungsi seperti yang diharapkan oleh pengguna [1]. Kumpulan fungsi dalam perangkat lunak yang memenuhi kebutuhan pengguna disebut fitur. Fitur menjadi suatu karakteristik atau ciri khas dari suatu perangkat lunak yang menjadi pembeda dari perangkat lunak satu dengan yang lainnya.

Pembuatan fitur baru merupakan hal yang tidak dapat dihindarkan karena akan membuat pengguna selalu merasa puas dan tertarik dalam menggunakan sistem. Hal tersebut sesuai dengan pendapat Kotler dan Armstrong yang menyatakan bahwa konsumen akan semakin terpuaskan dengan produk-produk yang sesuai dengan kebutuhannya [2]. Dengan selalu adanya permintaan fitur-fitur baru telah menciptakan masalah tersendiri bagi tim pengembang perangkat lunak. Fitur-fitur tersebut tidak dapat dipilih secara sembarangan. Tim pengembang harus memilih fitur mana yang mendapatkan prioritas rilis terlebih dulu yang sekiranya bisa diselesaikan tepat waktu, sesuai anggaran, dan bernilai tinggi buat pengguna. Pemilihan fitur-fitur ini sangat penting, karena ketidaktepatan pemilihan fitur-fitur berikutnya bisa berdampak pada berhentinya pengguna lama untuk menggunakan aplikasi. Bahkan, perusahaan bisa menanggung biaya yang lebih besar bila ternyata pengerjaan fitur yang dipilih tidak selesai tepat waktu.

Mengembangkan fitur yang bagus biasanya akan menemui masalah seperti waktu, kompleksitas, dan lain-lain. Tidak mungkin untuk mengerjakan dan merilis semua fitur yang diusulkan oleh *stakeholder* secara bersamaan karena masalah waktu, dana, orang, dan sebagainya. Oleh karena itu hanya fitur baru hasil dari perhitungan yang tepat yang dapat dimasukkan ke dalam sistem. Apalagi, del Sagrado et al., [3]

pernah menyatakan bahwa pemilihan fitur-fitur baru perangkat lunak merupakan tugas yang kompleks pada aktivitas pengembangan perangkat lunak.

Menurut Karl Wieggers dan Joy Beatty, metode yang biasanya digunakan untuk menyelesaikan masalah penentuan prioritas yaitu metode *In or out*, metode perbandingan dan pengurutan rank, metode *three-level scale*, metode MoSCoW, dan metode \$100 [4]. Metode *In or out* merupakan metode yang sederhana dimana keputusan diambil secara biner, yaitu suatu fitur akan diambil atau tidak berdasarkan keputusan *stakeholder* dalam menyusun daftar fitur. Metode perbandingan dan pengurutan rank melibatkan penetapan nomer urut prioritas yang unik untuk fitur, dengan mengurutkan peringkat dan melakukan perbandingan secara berpasangan untuk semua fitur maka akan teridentifikasi pasangan fitur yang memiliki prioritas tinggi. Metode yang termasuk *three-level scale* membagi menjadi tiga kategori yaitu High-priority untuk pengguna yang membutuhkan fitur dan pengguna menginginkan perlisian fitur di pembaruan berikutnya, Medium-priority jika pengguna membutuhkan fitur namun tidak terlalu membutuhkan di pembaruan berikutnya, dan Low-priority jika pengguna dapat menggunakan sistem tanpa perlu fitur baru serta pengguna dapat menunggu fitur baru di pembaruan sistem. Metode MoSCoW terbagi menjadi empat tingkat yaitu, Must mencakup persyaratan yang harus dipenuhi agar solusi dianggap berhasil, Should mengacu pada persyaratan yang penting dan harus disertakan dalam solusi jika mungkin namun tidak harus sukses, Could menunjukkan fitur yang dibutuhkan namun dapat ditunda perlisannya, sedangkan Won't mencakup fitur yang tidak ditetapkan untuk sekarang namun dapat disertakan untuk rilis mendatang. Metode \$100 adalah metode yang memberikan sumber dana terbatas sebesar 100 dolar dalam bentuk imajiner, dengan tujuan mencapai manfaat maksimal dari investasi pada suatu fitur. Menentukan kapan sebuah fitur harus dirilis tidak hanya didasarkan pada keputusan informal dan keputusan *stakeholder* semata, tetapi juga melibatkan teknik analitis dan matematis, seperti yang dikemukakan oleh Karl Wieggers dan Joy Beatty yang mengusulkan pendekatan penentuan prioritas berdasarkan nilai, biaya, dan risiko [4]. Metode ini menghubungkan nilai konsumen dengan fitur yang

diusulkan yang disebut QFD (*Quality Function Deployment*). Nilai, biaya, resiko akan dihitung dalam metode ini karena daya tarik dari fitur akan berbanding lurus dengan nilai yang diberikan dan berbanding terbalik dengan biaya serta resiko yang dihadapi. Jika fitur mempunyai resiko tinggi namun disesuaikan dengan rasio dari nilai/biaya maka akan mendapat prioritas yang tinggi.

Selain 5 metode tersebut, ada juga metode atau teknik optimasi. Metode ini menggunakan komputasional untuk mendapatkan solusi dengan cara menemukan nilai minimum serta nilai maksimum yang didapatkan dari fungsi objektif [5]. Solusi yang diperoleh akan digunakan untuk mendapatkan hasil terbaik dari setiap kasus yang dicari. Contoh kasus yang menggunakan teknik optimasi adalah pencarian rute terpendek dengan algoritma genetika, optimasi penentuan jadwal mata kuliah menggunakan algoritma genetika, optimasi penentuan vendor material pesawat menggunakan algoritma PSO (*Particle Swarm Optimization*), optimasi waktu penjadwalan produksi dengan algoritma CDS (*Campbell Dudek Smith*) dan lain sebagainya.

Kelebihan dalam menggunakan metode perbandingan dan pengurutan rank adalah cepat dalam memutuskan perilsan fitur yang dibutuhkan konsumen karena hanya memilah fitur menjadi sebuah skala yang ditentukan dengan metode tertentu. Kelemahan metode perbandingan dan pengurutan rank adalah penentuan perilsan fitur dilakukan jika terdapat sebuah prioritas yang seharusnya menjadi prioritas tertinggi namun menjadi prioritas lebih rendah karena metode ini bergantung pada kebutuhan konsumen.

Kelebihan dari penggunaan metode QFD adalah sistem matematisnya untuk menghitung prioritas tertinggi secara tepat berdasarkan variabel nilai, biaya, dan resiko. Kelemahan metode ini adalah jika terjadi kesalahan pada penentuan variabel maka akan membawa dampak yang besar untuk penentuan prioritas, karena variabel akan ditentukan oleh *stakeholder* yang mempunyai pemikiran yang berbeda-beda terhadap fitur yang ingin dirilis.

Penelitian ini menggunakan algoritma "*Simulated Annealing*" yang merupakan sebuah teknik metaheuristik yang digunakan untuk membantu menentukan fitur secara maksimal. Menurut penelitian

dari Universitas Almeria permasalahan rilis fitur akan lebih optimal jika menggunakan teknik metaheuristik [3]. Algoritma "*Simulated Annealing*" ini digunakan untuk mensimulasikan fitur-fitur mana saja yang harus terlebih dahulu dikerjakan.

Menurut penelitian Universitas Guadalajara dari Mexico, algoritma *Simulated Annealing* merupakan algoritma yang menggunakan skema berulang untuk agen pencarian mengubah posisi setiap iterasi dengan mempertimbangkan perubahan besarnya suhu [6]. Pada suhu tinggi pada algoritma *Simulated Annealing* dimungkinkan untuk bergerak jauh dalam ruang pencarian serta meningkatkan probabilitas penerimaan solusi yang tidak meningkat dari sebelumnya yang lebih tinggi. Sebaliknya, di suhu yang rendah kemungkinan untuk menerima solusi yang lebih buruk akan berkurang.

Penelitian ini bermaksud untuk menemukan sekumpulan tugas atau fitur yang memiliki usaha minimum dan kepuasan pengguna maksimum menggunakan algoritma *Simulated Annealing*. Kumpulan task atau fitur terpilih tersebut akan dimasukkan ke jadwal perilisan fitur baru berikutnya.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas maka rumusan masalah pada penelitian ini adalah bagaimana mengoptimalkan pemilihan tugas atau fitur yang menghasilkan usaha minimum dan kepuasan pengguna maksimum agar bisa dijadwalkan pada rilis berikutnya?

1.3 Tujuan Penelitian

Penelitian ini bertujuan sebagai berikut :

1. Merancang dan membangun program yang mengimplementasikan algoritma *Simulated Annealing* untuk menentukan fitur baru dengan usaha minimum dan kepuasan pengguna maksimum.

2. Mengevaluasi performa algoritma *simulated annealing* dengan *setting* parameter yang berbeda-beda.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberi manfaat sebagai berikut :

1. Membantu pengembang untuk mengotimalkan pengambilan keputusan terkait penentuan fitur baru dengan usaha minimum dan kepuasan pengguna maksimum.
2. Dapat digunakan sebagai referensi penelitian berikutnya tentang NRP (*Next Release Problem*).