BAB I
PENDAHULUAN

I.1 Latar Belakang Masalah

Masalah optimasi merupakan hal yang sering kita jumpai dalam pekerjaan sehari-hari, seperti optimasi pemilihan proyek, penjadwalan dan lain-lain. Menyelesaikan masalah optimasi dapat dilakukan dengan berbagai macam strategi, diantaranya adalah algoritma *greedy*, *bruteforce*, *dynamic programming* maupun genetika. Strategi algoritma *greedy* adalah yang paling populer diantara strategi algoritma yang lainnya, dikarenakan kesederhanaannya dan kemudahan penerapannya. Strategi algoritma *greedy* sering digunakan karena berguna untuk menghasilkan solusi yang menghampiri (*approximation*) optimum, dari pada menggunakan algoritma yang lebih rumit untuk menghasilkan solusi yang eksak [**1**].

Salah satu algoritma yang mengadopsi strategi algoritma *greedy* untuk diterapkan dalam penyelesaian kasus optimasi adalah algoritma *knapsack*. *Knapsack* dapat diilustrasikan seperti cara memasukkan beberapa barang ke dalam suatu kantong (*knapsack*). Kapasitas kantong ini memiliki keterbatasan ruang sehingga barang yang dimasukkan kedalam kantong tidak melebihi batas dan diperoleh keuntungan yang optimal. David Pisinger dalam *Alghoritm for Knapsack Problems* menyebutkan beberapa contoh penerapan kasus yang dapat diselesaikan dengan algoritma knapsack antara lain adalah masalah dalam muat barang kargo, pemotongan stok barang, pengkontrolan budget dan manajemen keuangan [**3**].

Hasil studi literatur tentang penelitian algoritma knapsack mengindikasikan, dari beberapa contoh kasus diselesaikan dengan algoritma *knapsack* ditemukan kesamaan karateristik parameter dengan kasus pemilihan proyek yang dimiliki oleh PT. GITS Indonesia. Pemilihan proyek yang dilakukan oleh PT. GITS dilakukan oleh seorang kepala kepegawaian selaku *analist project* (penentu proyek-proyek yang masuk ke perusahaan). Proyek-proyek ini kemudian dipilih oleh analis dengan pertimbangan-pertimbangan tertentu. Beberapa hal yang menjadi penentu pertimbangan antara lain *profit* dan *resources* (jumlah sumber

daya manusia). Jumlah sumber daya manusia yang terdapat di perusahaan ini dapat diibaratkan dengan kantong (*knapsack*) yang memiliki keterbatasan jumlah. *Profit* dan *resources* atau sumber daya manusia dalam kasus pemilihan proyek di PT. GITS Indonesia ini akan menjadi parameter dalam analisis algoritma *knapsack* yang dilakukan. Kemiripan karateristik kasus ini akan menjadi lebih baik ketika algoritma yang digunakan dianalisis performansinya terlebih dahulu, sehingga keoptimalannya dapat dibuktikan.

Analisis performansi algoritma dilakukan agar diketahui efisiensi dan kelayakan algoritma pada kasus yang sedang diuji. Tanpa dilakukannya analisis algoritma maka akan terjadi masalah terhadap pengimplementasian algoritma pada kebutuhan dan masalah yang dihadapi. Algoritma yang akan digunakan tidak harus mewah atau canggih namun menggunakan sebuah algoritma sebaiknya tepat dan cocok pada kasus yang sedang diuji. Sebagai contoh jika kita membutuhkan algoritma yang tepat untuk diterapkan dalam kasus pemilihan proyek maka algoritma yang dapat diterapkan dengan baik adalah algoritma *knapsack*.

Berdasarkan dari permasalahan yang telah dipaparkan pada paragraf sebelumnya maka diperlukan analisis performansi algoritma *knapsack* terhadap kasus pemilihan proyek. Penerapan algoritma terhadap perangkat lunak merupakan salah satu cara yang dapat digunakan untuk menganalisis dan mengetahui performansi algoritma *knapsack* terhadap kasus pemilihan proyek ini.

I.2 Perumusan Masalah

Berdasarkan latar belakang masalah maka perumusan masalah pada penelitian ini adalah bagaimana mengoptimalkan hasil pemilihan proyek dengan menganalisis performansi algoritma *Knapsack*.

I.3 Maksud dan Tujuan

Maksud dari penelitian skripsi ini adalah untuk melakukan analisis performansi algoritma *knapsack* pada kasus pemilihan proyek.

Adapun tujuan dari pembuatan skripsi ini adalah :

1. Mengetahui performansi algoritma *knapsack* pada kasus pemilihan proyek di PT. GITS Indonesia.

2. Mendapatkan hasil yang optimal dari kasus pemilihan proyek menggunakan strategi algoritma *greedy by density* dengan jenis algoritma *fractional knapsack*.

I.4 Batasan Masalah

Batasan masalah yang ditentukan dalam penelitian ini adalah sebagai berikut :

1. Strategi algoritma yang digunakan untuk menyelesaikan permasalahan yang sedang diuji menggunakan stretegi algoritma *greedy* dengan pendekatan *greedy by density*.
2. Jenis algoritma yang dianalisis adalah *fractional Knapsack*.
3. Model analisis yang digunakan adalah model kompleksitas waktu (Tn).
4. Notasi asimtotik yang digunakan adalah O (big oh).
5. Sumber data proyek yang akan diuji berasal dari PT. GITS Indonesia.
6. Simulasi yang dilakukan untuk pengujian performansi algoritma *knapsack* menggunakan perangkat lunak berbasis desktop.

I.5 Metodologi Penelitian

Metodologi penelitian yang akan digunakan dalam pembuatan skripsi ini menggunakan metodologi Analisis deskriptif. Analisis deskriptif adalah metode analisis dengan mendeskripsikan atau menggambarkan data yang telah terkumpul sebagaimana adanya tanpa membuat kesimpulan yang berlaku untuk umum. Metodologi ini terbagi menjadi dua metode yaitu metode pengumpulan data dan metode pembangunan perangkat lunak.

I.5.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan terdiri dari dua cara, diantaranya :

1. Studi literatur

   Studi literatur adalah metode pengumpulan data dengan cara mengumpulkan pustaka yang menunjang penelitian yang akan dikerjakan. Pustaka tersebut dapat berupa e-book, buku, artikel, jurnal, laporan akhir, dan sebagainya.

2. Studi Lapangan

Studi lapangan adalah metode pengumpulan data dengan cara berinteraksi langsung dengan orang-orang yang berada di lingkungan tempat studi kasus. Studi lapangan yang dilakukan adalah dengan melakukan wawancara atau tanya jawab dengan kepala kepegawaian.

I.5.2 Metode Pembangunan Perangkat Lunak

Pengembangan perangkat lunak dalam penelitian ini mengadopsi metode pengembangan *classic life style* atau *waterfall*. Metode pembangunan perangkat lunak *waterfall* dapat dilihat pada Gambar I-1. Tahapan yang ada pada metode pengembangan *waterfall* adalah sebagai berikut :

a. *Requirements analysis and definition*

Tahap *Requirements analysis and definition* merupakan tahap pengumpulan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

b. *System and software design*

Tahap *System and software design* merupakan tahap mendesain perangkat lunak yang dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

c. *Implementation and unit testing*

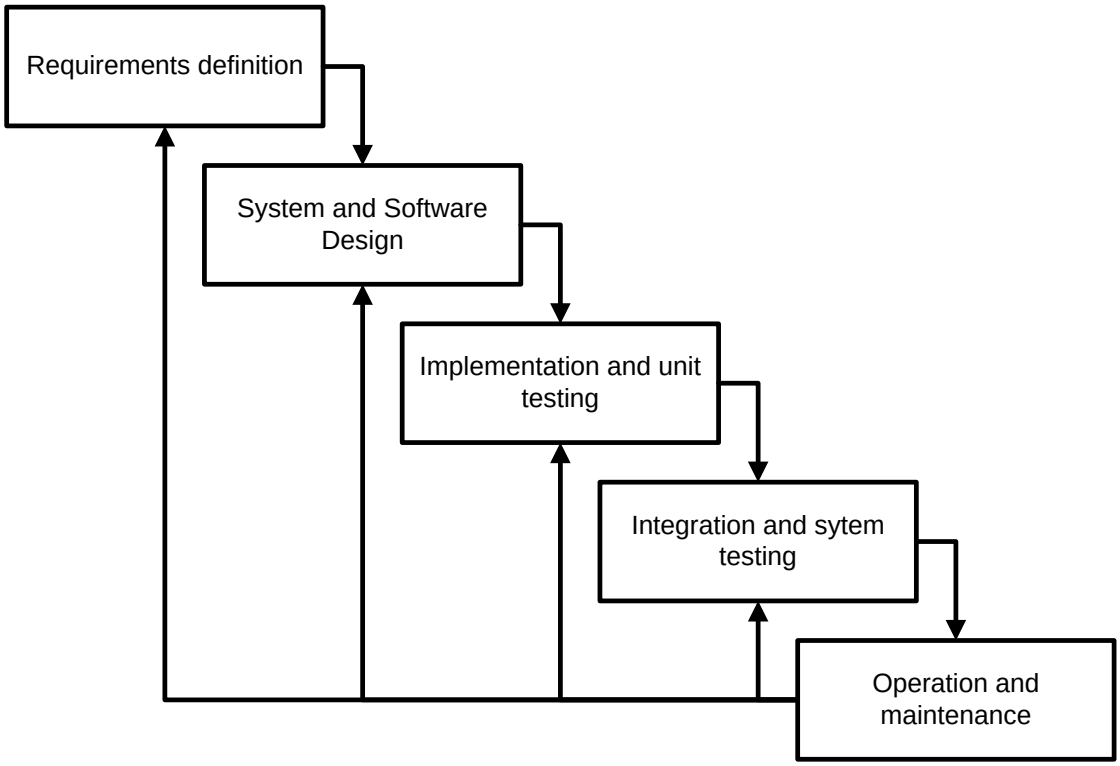
Tahap *Implementation and unit testing* merupakan tahap hasil desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

d. *Integration and system testing*

Tahap *Integration and system testing* merupakan tahap penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).

e. *Operation and maintenance*

Tahap *Operation and maintenance* merupakan tahap mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.



Gambar I-1 Model Proses Waterfal [3**]**

I.6 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Adapun sistematika penulisan skripsi ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab I merupakan proses yang menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan maksud dan tujuan penelitian, yang kemudian diikuti dengan pembatasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab II membahas tentang berbagai konsep dasar dan teori-teori yang berkaitan dengan topik yang diangkat dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang pernah dilakukan sebelumnya termasuk sintesisnya.

BAB III ANALISIS ALGORITMA

Bab III berisi tentang analisis algoritma, pendekatan strategi algoritma, dan kebutuhan untuk alat simulasi berupa perangkat lunak yang dibangun sesuai dengan metode pengembangan perangkat lunak yang digunakan.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab 4 berisi hasil implementasi analisis dan perancangan simulasi yang dilakukan, serta hasil pengujian simulasi untuk mengetahui apakah aplikasi yang dibangun sudah memenuhi kebutuhan. Pengujian yang diakukan dalam penelitian ini adalah dengan menggunakan *white box testing* dan *black box testing*.

BAB V KESIMPULAN DAN SARAN

Bab V berisi kesimpulan hasil pengujian dari perangkat lunak yang telah dibangun berdasarkan tujuan yang ingin dicapai dan saran yang dapat diberikan untuk perangkat lunak ini untuk kemudian dapat dikembangkan kembali.