

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang Masalah

Sebagai alat transportasi yang praktis dan murah, motor menjadi pilihan utama bagi setiap orang. Hal ini dapat dilihat dari jumlah kendaraan bermotor khususnya sepeda motor yang berada di jalan. Dengan angka pertumbuhan rata-rata jumlah sepeda motor di Indonesia yang masih berada pada angka 8 - 10% per tahun, maka sepeda motor masih akan terus mendominasi di jalanan [1].

Penambahan performa dilakukan jika performa sepeda motor tersebut tidak menunjang kebutuhan pengendara, seperti untuk *touring* yang membutuhkan akselerasi tinggi agar tidak tertinggal dari rombongan. Banyak hal yang dapat dilakukan oleh pengendara untuk menambah performa kendaraan mulai dari penyuplai bahan bakar, perubahan dapur pacu, hingga perubahan rasio transmisi dari sepeda motor. Modifikasi mesin sepeda motor merupakan hal yang rumit dan kompleks. Penentuan *sparepart* dilakukan berdasarkan perhitungan spesifikasi *sparepart* dengan spesifikasi *sparepart* lain yang berkaitan agar modifikasi menjadi optimal. Dampaknya jika salah dalam menentukan *sparepart*, bukan hanya performa mesin yang lebih buruk namun juga akan menimbulkan biaya yang lebih besar. Sehingga diperlukan perhitungan yang matang terhadap *sparepart* yang akan dipilih sebelum dipasang pada sepeda motor [2].

Untuk mendapatkan hasil modifikasi yang optimal, dapat menggunakan beberapa pendekatan dari algoritma untuk optimasi seperti algoritma *Greedy*,

*Brute Force*, *Artificial Bee Colony*, *Cheapest Insertion Heuristics*, *Clark Weight Saving*, Simpleks Direvisi (Primal), dan Genetika. Algoritma *greedy* menunjukkan performa paling cepat dibandingkan dengan algoritma yang lain. Sedangkan berdasarkan hasil yang didapat tidak ada algoritma yang menunjukkan performa yang stabil, setiap algoritma menunjukkan hasil optimum untuk kasus yang berbeda. Maka tidak ada satupun algoritma yang berlaku umum dan bisa digunakan untuk menyelesaikan semua jenis masalah [3]. Selain karena kesederhanaan dan kemudahan dalam penerapannya, algoritma *greedy* bisa menjadi pilihan karena kemangkusannya. Dibandingkan dengan algoritma *clark weight saving*, algoritma *greedy* menunjukkan hasil yang lebih optimum untuk *vertex* yang lebih banyak [4]. Bahkan algoritma *greedy* menunjukkan hasil yang lebih baik dari algoritma simpleks direvisi (primal) dalam aspek performa, jumlah *vertex* yang bisa diproses, dan kelayakan dari himpunan solusi [5]. Serta dalam perbandingan algoritma *greedy* dengan algoritma *brute force* yang diterapkan dalam pencarian koin menunjukkan bahwa algoritma *greedy* dapat menemukan koin dalam kondisi apapun dengan jumlah percobaan yang lebih sedikit [6].

Himpunan solusi merupakan solusi paling optimal dari setiap himpunan kandidat. Setiap himpunan solusi pada setiap langkah dalam algoritma *greedy* disebut dengan optimum lokal. Algoritma *greedy* mencari solusi optimum untuk setiap langkah sehingga setiap solusi pada setiap langkah merupakan himpunan optimum lokal yang mengarah pada optimum global ketika setiap langkah sudah dilewati. Sehingga algoritma *greedy* mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global [7]. Langkah-langkah tersebut memiliki

karakteristik yang mirip dalam langkah modifikasi. Modifikasi dilakukan dengan mengubah *piston* berdasarkan panjang dari *stroke*. Kemudian mengubah ukuran *valve* berdasarkan diameter *piston*. Dan terus bersinambung hingga perubahan pada penyuplai bahan bakar.

Dalam penerapannya setiap pilihan spesifikasi *sparepart* dijadikan sebagai himpunan kandidat untuk setiap jenis *sparepart*. Sehingga jika sebuah *sparepart* diganti dengan spesifikasi yang paling optimal diantara setiap kandidat, maka akan mengarah kepada modifikasi yang paling optimal. Jadi sebelum dilakukan percobaan pemilihan *sparepart* secara nyata, dilakukan simulasi pemilihan dengan mencoba setiap *sparepart* dan menentukan *sparepart* mana yang dipilih untuk diterapkan agar lebih optimal namun tetap ekonomis. Jika tidak, bukan hanya performa buruk yang menjadi dampak dari ketidakmatangan persiapan modifikasi, juga angka kecelakaan sepeda motor akibat motor modifikasi akan semakin tinggi. Sehingga diharapkan hal ini dapat mengurangi biaya dan waktu operasional dalam modifikasi sepeda motor serta menekan angka kecelakaan akibat buruknya modifikasi.

Berdasarkan uraian di atas, algoritma *greedy* akan diterapkan dalam aplikasi yang dapat mensimulasikan perhitungan performa sebelum dilakukan modifikasi secara nyata melalui perangkat *smartphone* android. Tugas akhir ini diarahkan pada judul **“Implementasi Algoritma Greedy pada Aplikasi Pemilihan Sparepart Modifikasi Sepeda Motor berbasis Android.”**

## 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka dirumuskan masalah-malah pokok dalam penelitian ini di antaranya.

- a. Bagaimana algoritma *greedy* menentukan *sparepart* modifikasi yang optimal?
- b. Bagaimana kelayakan himpunan solusi yang dihasilkan oleh algoritma *greedy* dalam menentukan *sparepart* modifikasi?

## 1.3 Tujuan dan Manfaat

Tujuan dari pembuatan aplikasi untuk optimasi pemilihan *sparepart* dalam modifikasi sepeda motor ini adalah.

- a. Menerapkan algoritma *greedy* untuk menentukan *sparepart* dalam modifikasi sepeda motor.
- b. Memberikan arahan dan masukkan dalam menentukan *sparepart* untuk hasil yang optimal dalam modifikasi sepeda motor.

## 1.4 Batasan Masalah

Batasan masalah dari pembuatan aplikasi untuk optimasi pemilihan *sparepart* dalam modifikasi sepeda motor ini adalah.

- a. Kriteria sepeda motor yang digunakan hanya sepeda motor empat langkah *non matic* dan *non fuel injection* yang dipasarkan di Indonesia.
- b. Pengguna melakukan *input* nominal *budget* yang tersedia sebagai batasan dari total harga *sparepart* dalam modifikasi.

- c. Terdapat asas ketergantungan dalam pemilihan *sparepart*.
- d. Kriteria *sparepart* yang digunakan ialah 4 jenis *sparepart* mulai dari penyuplai hingga saluran pembuangan.
- e. Perancangan sistem menggunakan UML yang terdiri dari empat diagram yaitu *Class Diagram*, *Use Case Diagram*, *Sequence Diagram*, dan *Activity Diagram* dengan aplikasi StarUML v2.7.1.
- f. Aplikasi dibangun menggunakan bahasa *java* dengan *Android Studio* 1.3.
- g. *Database* dibangun menggunakan SQLite.
- h. Aplikasi hanya dapat berjalan pada Android dengan versi minimal 2.3.3 *Gingerbread*.

## 1.5 Metodologi Penelitian

Metodologi yang digunakan dalam melakukan penelitian ini terbagi dalam dua tahap.

### 1.5.1 Teknik Pengumpulan Data

- a. Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul skripsi.

- b. Observasi

Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

### 1.5.2 Pendekatan Sistem

Pemrograman berorientasi objek (*Object Oriented Programming*) merupakan suatu bentuk pemrograman yang memodelkan masalah dengan pendekatan objek. Proses perancangan pada pemrograman berorientasi objek menggunakan Diagram *Unified Modelling Language* (UML). Terdapat empat komponen UML yang akan digunakan dalam perancangan aplikasi, di antaranya.

a. *Use Case Diagram*

Diagram ini digunakan untuk merekam persyaratan fungsional dari sebuah sistem. *Use case* mendeskripsikan interaksi dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

b. *Sequence Diagram*

Diagram ini menjelaskan bagaimana setiap objek saling berkolaborasi dalam setiap operasinya. Sebuah *sequence* menjabarkan *behavior* dari sebuah scenario tunggal, sehingga dapat menunjukkan pesan objek di dalam *use case*.

c. *Class Diagram*

Diagram ini mendeskripsikan jenis-jenis objek dalam sistem serta relasi statis yang ada di antara objek-objek tersebut. Dalam *class diagram* sebuah objek ditulis dengan properti, operasi, dan jenis relasinya.

d. *Activity Diagram*

Diagram ini menggambarkan logika prosedural, proses bisnis, dan alur kerja. Sehingga diagram ini memiliki peran yang mirip dengan diagram alir. Namun pada diagram ini mendukung *behavior* paralel [8].

### 1.5.3 Pengembangan Sistem

Model pengembangan yang digunakan adalah model proses *Prototype*. Dengan Metode ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Model pengembangan ini dapat dilihat pada Gambar 1.1.



Gambar 1.1 Model Pengembangan *Prototype*

a. Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

b. Merancang dan Membuat *Prototype*

Pada tahap ini, dilakukan perancangan dan pembuatan *prototype* sistem. *Prototype* yang dibuat disesuaikan dengan kebutuhan system yang telah didefinisikan sebelumnya dari keluhan pelanggan.

c. Uji coba

Pada tahap ini, *Prototype* dari system diuji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki *Prototype* yang ada [9].

## 1.6 Sistematika Penulisan

Sistematika penulisan dalam penyusunan skripsi ini berdasarkan buku Pedoman Skripsi Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Gunung Djati Bandung tahun 2015, terbagi menjadi beberapa bab yang akan dibahas sebagai berikut.

### BAB I PENDAHULUAN

Menjelaskan secara umum mengenai Latar Belakang Masalah, Perumusan Masalah, Tujuan dan Manfaat, Batasan Masalah, Metodologi Penelitian, dan Sistematika Penulisan yang digunakan dalam Skripsi ini.

### BAB II STUDI PUSTAKA

Berisi teori, landasan, paradigma, cara pandang, metode-metode yang telah ada dan atau yang digunakan dalam Aplikasi serta konsep yang telah diuji kebenarannya.

### BAB III PERANCANGAN SISTEM

Pada bagian ini akan dibahas mengenai pengembangan sistem atau pembuatan sistem baru dengan disertai model perancangan secara lengkap.

### BAB IV IMPLEMENTASI SISTEM



Menjelaskan implementasi dari perancangan yang telah dibuat dan pembahasannya, dengan disertai tampilan dari aplikasi serta modul aplikasi yang mendukungnya.

#### BAB V PENUTUP

Berisi simpulan yang merangkum seluruh isi dari skripsi yang telah dibahas. Selanjutnya akan dikemukakan saran-saran mengenai perluasan, pengembangan, pendalaman, dan pengkajian ulang dari penulis bagi pengembang aplikasi untuk masa yang akan datang [10].





uin

UNIVERSITAS ISLAM NEGERI  
SUNAN GUNUNG DJATI  
BANDUNG