BAB I
PENDAHULUAN

I.1. Latar Belakang

Algoritma adalah prosedur komputasi yang terdefinisi dengan baik yang menggunakan beberapa nilai sebagai masukan dan menghasilkan beberapa nilai yang disebut keluaran. Jadi, algoritma adalah deretan langkah komputasi yang mentransformasikan masukan menjadi keluaran **[1]**. Pencarian adalah proses yang fundamental dalam pengolahan data. Jadi, Algoritma pencarian adalah sebuah algoritma yang menerima masukan berupa sebuah masalah dan menghasilkan sebuah solusi untuk masalah tersebut, yang biasanya didapat dari evaluasi beberapa kemungkinan solusi dalam proses pengolahan data **[2]**. Sedangkan Algoritma pencarian *string* atau sering disebut juga pencocokan *string* adalah algoritma untuk melakukan pencarian semua kemunculan string pendek *pattern* 0..n-1 yang disebut *pattern* di *string* yang lebih panjang teks 0..m-1 disebut teks **[3]**.

Ada sekitar 35 algoritma pencarian kata yang bisa digunakan dalam perangkat baik merupakan algoritma yang diciptakan dari awal maupun berupa pengembangan dari algoritma yang sudah ada. Dua di antaranya yaitu algoritma *Karp-Rabin* dan algoritma *Reverse Colussi*. Algoritma *Karp-Rabin* adalah algoritma pencarian *string matching* yang arah pencariannya dari kiri ke kanan dan pencariannya menggunakan fungsi *Hasing* untuk membantu mempercepat pencarian *string* **[4]**. Berdasarkan referensi jurnal *"Perbandingan Algoritma Karp-Rabin dengan Algoritma Knuth-Morris-Pratt dalam pencarian string"* dari Hadi Saloko (ITB) yang menyatakan bahwa algoritma *Karp-Rabin* lebih unggul dibandingkan dengan algoritma *Knuth-Morris-Pratt* dalam kasus pencarian *string* dengan *pattern* yang panjang dan kasus pencarian *string* dengan *multipattern* **[5]**. Sedangkan Algoritma *Reverse Colussi* yaitu pencarian yang telah ditentukan secara spesifik oleh sistem atau algoritma tersebut dimana algoritma *Reverse Colussi* merupakan perbaikan dari algoritma *Boyer-Moore* yang merupakan pencarian dari kanan ke kiri. Algoritma *Reverse Colussi* terdiri dari 2 fase

pencarian, yaitu fase pemrosesan awal dan fase pencarian. Pada fase pemrosesan awal, dilakukan pencarian sepasang karakter untuk menentukan posisi dan pergeseran *pattern*. Pada fase pencarian, dilakukan operasi pencocokan *pattern* terhadap teks atau gambar. Berdasarkan jurnal "P*erbandingan Algoritma Reverse Colussi, Knuth-Morri-Pratt, Boyer-Moore, dan Colussi dalam pencarian string"* dari Didik Haryadi (ITB) menyatakan bahwa algoritma *Reverse Colussi* lebih cepat dalam pencarian *string* **[6]**.

Masing-masing Algoritma pencarian memiliki keunggulan dan kekurangan dalam pencarian, pencocokan, kecepatan, ketepatan, dan lain-lain. Dengan melakukan analisis perbandingan performansi maka akan dapat diketahui kecepatan dan ketepatan dalam waktu proses pencarian dan pencocokan kata dan persentase *memory* yang dipakai dalam satu kali proses dari masing - masing algoritma tersebut.

Dari permasalahan yang dipaparkan maka untuk menentukan performansi dari kedua algoritma dalam mencari dan mencocokkan kata, kedua Algoritma tersebut akan diimplementasikan pada sebuah *Word Game* untuk menentukan mana algoritma yang lebih optimal berdasarkan hasil perbandingan performansi kedua algoritma dari kecepatan dan ketepatan waktu proses dan persentase penggunaan *memory* dalam 1 kali proses.

I.2. Rumusan Masalah

Berdasarkan uraian dari latar belakang masalah, maka rumusan masalahnya adalah :

1. Bagaimana membandingkan algoritma *Reverse Colussi* dengan algoritma *Karp-Rabin* ?
2. Algoritma apa yang lebih optimal antara kedua algoritma dalam mencari dan mencocokkan kata ?

I.3. Maksud dan Tujuan

Berdasarkan masalah yang telah dikemukakan diatas, maka maksud dari penulisan tugas akhir ini yaitu untuk mengetahui mana algoritma yang lebih

optimal dengan cara membandingkan algoritma *Reverse Colussi* dengan algoritma *Karp-Rabin* dari sisi performansi dalam mencari dan mencocokkan *string*.
Adapun tujuan penulisan tugas akhir ini antara lain :

1. Membandingkan kedua Algoritma dengan menentukan performansi kecepatan dan ketepatan waktu proses dan persentase *memory* yang digunakan pada satu kali proses dalam mencari dan mencocokkan kata.
2. Untuk mengetahui mana algoritma yang lebih optimal antara kedua algoritma dalam mencari dan mencocokkan kata berdasarkan hasil performansi yang didapat.

I.4. Batasan Masalah

Dalam pembangunan *game* ini, perlu diberikan beberapa batasan masalah agar mempermudah saat melakukan perancangan. Berikut batasan masalahnya :

1. *Game* ini dibuat untuk membandingkan performansi kedua algoritma dari kecepatan dan ketepatan waktu proses dan persentase *memory* yang digunakan dalam satu kali proses.
2. Pendekatan analisis pembangunan perangkat lunak menggunakan pendekatan analisis berorientasi objek.
3. *Output* pencarian dari perangkat lunak adalah kata yang di cari, kecepatan waktu proses dan persentase *memory*.
4. kata yang dicari memiliki relevansi dengan data *string* yang ada di *wordlist*.
5. Jenis game berupa *Word Game*.
6. Pemodelan perancangan sistem menggunakan diagram UML.

I.5. Metodologi Penelitian

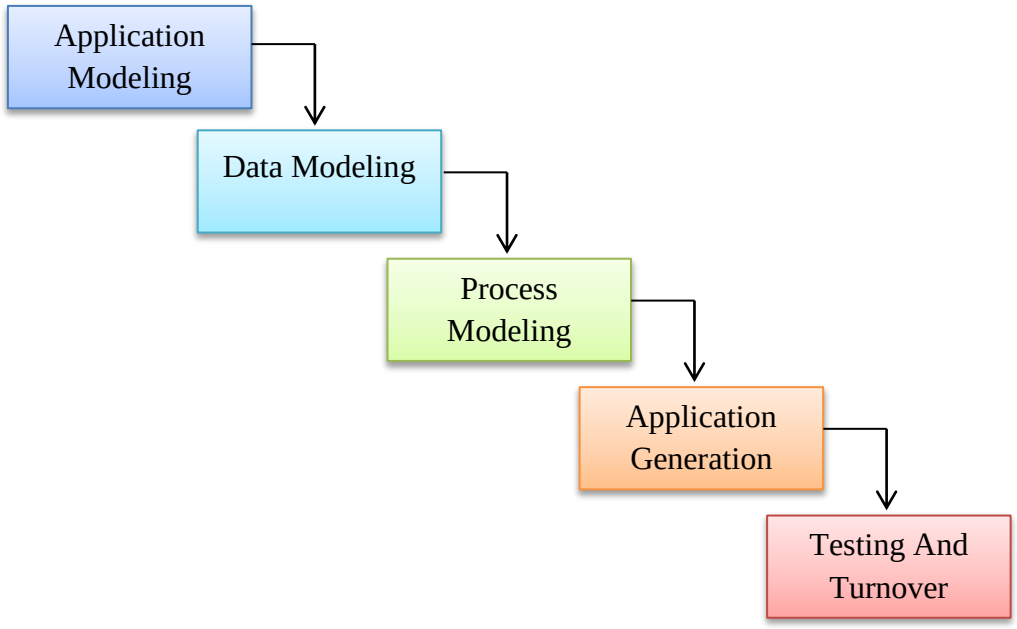
Metodologi penelitian yang akan digunakan dalam pembuatan Skripsi ini menggunakan metodologi Analisis Deskriptif, yaitu metode penelitian menggunakan studi kasus. Metode ini dibagi menjadi 2 tahap, yaitu sebagai berikut :

1. Tahap pengumpulan data

Pada tahap ini data dikumpulkan berdasarkan studi literatur berupa jurnal *"Perbandingan string dengan algoritma reverse colussi*" yang membahas tentang proses pencarian dan pencocokkan kata dan membandingkan antara algoritma *Reverse Colussi, Colussi, Boyer-Moore*, dan *Knuth-Morris-Prat*. Dari 4 algoritma tersebut menyatakan algoritma *Reverse Colussi* lah yang tercepat dan akurat dalam mencari dan mencocokkan kata.

2. Tahap pengembangan perangkat lunak

Teknik analisis data dalam pembuatan perangkat lunak menggunakan model *Rapid Application Development* (RAD), yang meliputi beberapa proses diantaranya :



Gambar I.1. Model Rapid Application Development (RAD) [7].

1. Tahap Pemodelan Aplikasi (*Application Modeling*)

Pada tahap ini menjelaskan dan menentukan apa saja yang dibutuhkan dalam pembuatan aplikasi yaitu menentukan apa jenis game yang cocok untuk

mengimplementasikan perbandingan kedua algoritma pencarian dalam mencari dan mencocokkan kata.

2. Tahap Pemodelan Data (*Data Modeling*)

Pada tahap ini menggabungkan data dan informasi dari pemodelan aplikasi dan studi literatur yang menjadi acuan dalam perbandingan dan pembuatan aplikasi.

a. Informasi dari pemodelan aplikasi

informasi yang didapat dari pemodelan aplikasi dipilah-pilah menjadi sekumpulan objek.

b. Studi literatur

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

3. Tahap Pemodelan Proses (*Process Modeling*)

Pada tahap ini menganalisis informasi yang didapat dari jurnal yang berkaitan dengan kedua algoritma kemudian dibuat pemodelan UML dan tabel perbandingannya untuk mengetahui performansi dari kecepatan waktu proses dan persentase *memory*.

4. Tahap Pembuatan Aplikasi (*Application Generation*)

Pada tahap ini adalah tahap perancangan aplikasi untuk mengimplementasikan kedua algoritma dalam mencari dan mencocokkan kata dimana tujuan *output*nya adalah mencari perbandingan performansi kedua algoritma dari kecepatan waktu proses dan persentase *memory*.

5. Tahap Pengujian (*Testing And Turnover*)

Pada tahap ini adalah tahap terakhir dari proses pembuatan aplikasi yaitu menguji aplikasi yang telah siap digunakan untuk mendapatkan hasil mana algoritma yang lebih optimal dari perbandingan performansi antara kedua algoritma dalam mencari dan mencocokkan kata dari kecepatan dan ketepatan waktu proses dan persentase *memory* yang digunakan dalam satu kali proses.

I.6. Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Menguraikan tentang latar belakang, rumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tentang landasan teori berupa pengertian dari *word search game*, *string matching*, pengertian algoritma, teori pengujian algoritma, analisis kompleksitas, metode pembangunan perangkat lunak berupa pengertian (model RAD, *Artificial Intelligence*, OOP, UML, bahasa pemrograman, *Java Script*, *Code editor* dan *Browser*).

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini menganalisis masalah, analisis kasus algoritma, analisis kompleksitas, analisis kebutuhan perangkat lunak seperti membuat UML (*use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*), dan perancangan seperti perancangan antarmuka dan jaringan semantik.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi tentang perancangan sistem dan tahapan-tahapan yang dilakukan untuk menerapkan sistem yang telah dirancang, membuat tabel perbandingan dan menampilkan hasil pengujian aplikasi.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran yang sudah diperoleh dari hasil penulisan tugas akhir dan daftar pustaka.