

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam dunia akademik, karya tulis seseorang dinilai dari keasliannya. Karya tulis dalam era globalisasi bisa didapat dengan menggunakan internet. Akses terhadap internet untuk mencari konten dari sebuah karya tulis semakin mudah. Akibatnya, karya-karya tulis tersebut semakin banyak persalinan sehingga keaslian nya dipertanyakan, tentu ini mengacu terhadap pendeteksian kemiripan teks untuk karya tulis sehingga keasliannya dapat diketahui dengan tepat.

Dalam plagiarisme, ada beberapa cara yang dilakukan untuk mengatasi permasalahan ini, yaitu dengan mencegah dan mendeteksi. Mencegah berarti menjaga atau menghalangi agar plagiarisme tidak dilakukan. Mendeteksi berarti melakukan usaha untuk menentukan tindakan plagiat yang telah dilakukan.

Namun di sisi lain, seringkali terjadi plagiat terhadap konten karya tulis atau bahkan tugas yang di kumpul. Hal ini berujung pada kurang nya kreatifitas siswa dalam menuangkan ide dan pendapat pribadi ke dalam konten yang ingin disampaikan. Juga permasalahan seringkali terjadi dengan duplikasi hak cipta. Hal ini berdampak pada sisi siswa, dimana nanti ke depan dalam menuangkan pendapatnya akan sulit terutama ketika dimintai pendapat pribadi.

### **1.2 Rumusan Masalah**

Adapun beberapa masalah yang akan di bahas dalam laporan penelitian ini :

1. Bagaimana implementasi algoritma *Rabin Karp* ke dalam aplikasi plagiarisme?

2. Bagaimana membangun sistem pendeteksi plagiat dengan menggunakan algoritma Rabin Karp?

### 1.3 Tujuan dan Manfaat

Tujuan dari pembuatan aplikasi ini adalah untuk membangun aplikasi pendeteksi plagiarisme dari tugas dan makalah yang ada di ruang lingkup sekolah berdasarkan tingkat kesamaan kalimat untuk mengetahui seberapa besar kesamaan nya.

Manfaat dari penelitian ini adalah :

1. Mempermudah pengecekan dokumen dengan konten yang sama.
2. Antara dokumen yang asli dengan dokumen yang diuji, dapat ditentukan persentase kemiripan (*similarity*).
3. Dapat dijadikan sebagai bahan pertimbangan dalam menentukan plagiarisme.

### 1.4 Ruang Lingkup

Berdasarkan penelitian yang penulis lakukan, penulis akan membatasi beberapa hal, yaitu :

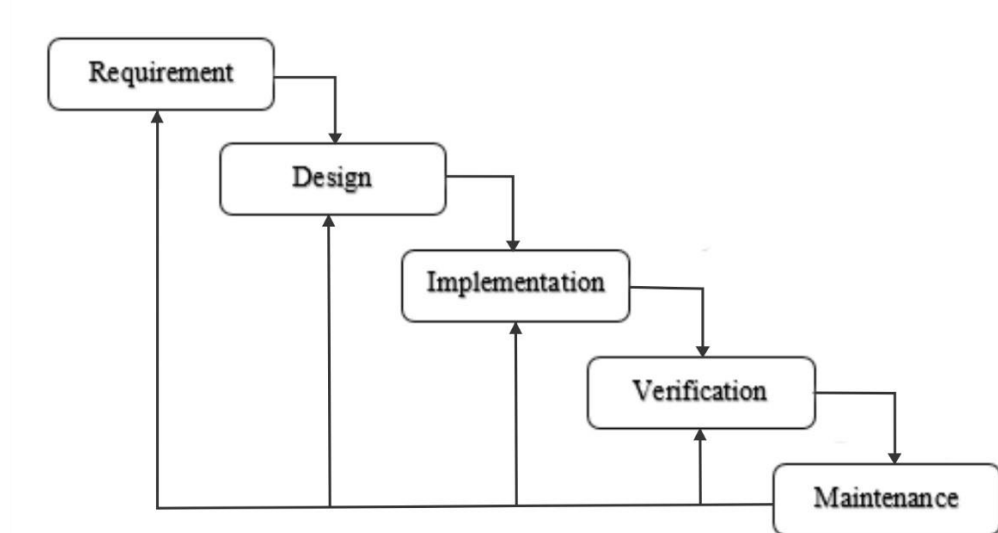
1. Dokumen yang dapat di deteksi adalah *file* teks digital yang berbentuk *.pdf*, *.doc* atau *.docx*, *.txt*.
2. Hanya menguji data berupa teks, tidak menguji data dalam bentuk gambar dan suara.
3. Hasil perbandingan dengan sistem pendeteksi plagiarisme yang dibuat tidak menunjukkan mutlak atau tidaknya terjadi plagiarisme antara dua dokumen teks. Hasil tersebut hanya sebagai panduan agar memberi perhatian lebih terhadap dua teks tersebut.

4. Sistem yang dibangun hanya sebagai alat bantu untuk menemukan kesamaan. Keputusan akhir tetap berada di tangan pengguna atau guru.

## 1.5 Metode Penelitian

### 1.5.1 Metode *Waterfall*

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan atau pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012). Tahapan metode *waterfall* dapat dilihat pada gambar di bawah ini :



Gambar 1.1 Metode *Waterfall* (Pressman, 2012)

Adapun tahap-tahap yang dijelaskan di bawah ini :

1. *Requirement Analysis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

## 2. *System Design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

## 3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

## 4. *Verification*

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

## 5. *Maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah

sebelumnya. Perbaiki implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

### **1.5.2 Metode *Interview***

Metode yang dilakukan dengan mewawancarai untuk memperoleh keterangan dalam penelitian dengan cara tanya-jawab sambil bertatap muka antara pewawancara dengan penjawab.

### **1.5.3 Studi Literatur**

Studi yang dilakukan dengan membaca dan mempelajari referensi sebagai pelengkap data yang sumbernya dari *internet* maupun yang lain.

## **1.6 Sistematika Penulisan**

Untuk memudahkan dalam mengikuti seluruh uraian dan pembahasan atas skripsi ini, maka penulisan skripsi ini dilakukan dengan sistematika berikut :

### **BAB 1 : PENDAHULUAN**

Bab ini berisikan tentang latar belakang masalah, rumusan masalah, tujuan dan manfaat dari penelitian, dan sistematika penulisan.

### **BAB 2 : TINJAUAN PUSTAKA**

Bab ini menjelaskan teori-teori yang akan dipakai dalam skripsi ini. Yang berisi pengertian dan kutipan buku atau internet yang berhubungan dengan penyusunan skripsi.

### **BAB 3 : ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan masalah yang akan dihadapi oleh skripsi ini beserta dengan metode yang akan dipakai seperti *Class diagram*, UML, dan sebagainya.

### **BAB 4 : IMPLEMENTASI**

Bab ini menjelaskan analisa dari sistem yang disusulkan dengan sistem algoritma yang dipakai, algoritma *flowchart*, serta *mindmap* yang dipakai.

## BAB 5 : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran yang berkaitan dengan analisa dan optimalisasi sistem berdasarkan hasil yang telah diuraikan pada bab-bab sebelumnya.