

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi pada era saat ini semakin pesat dan seringkali diterapkan dalam kehidupan sehari-hari, baik dalam dunia bisnis, industri, maupun kehidupan pribadi. Salah satu perkembangan teknologi yang sangat nampak adalah tersedianya banyak aplikasi yang digunakan oleh masyarakat luas. Dalam pembuatan aplikasi membutuhkan seseorang untuk membuat dan menulis kode program yang biasa disebut sebagai *programmer* [1]. Dalam penulisan kode program, *programmer* seringkali mengalami masalah. Salah satu masalah yang sering terjadi disebabkan karena adanya anomali pada kode program [2].

Anomali secara umum adalah suatu kejadian yang tidak sesuai dengan pola atau standar yang diharapkan dalam suatu sistem [3]. Anomali dapat terjadi dalam berbagai konteks, termasuk pengembangan perangkat lunak, statistik, data, dan kode program. Dalam konteks kode program sendiri, anomali adalah keanehan, atau ketidaksesuaian yang tidak lazim atau tidak diharapkan pada suatu kode program. Salah satu dari masalah yang diakibatkan karena adanya anomali yaitu adanya kode program yang memiliki fungsi berbahaya di dalamnya, seperti pada fungsi “*GET_ACCOUNTS*” yang merupakan salah satu *permission* di aplikasi android yang digunakan dalam mengakses data atau informasi dari akun atau kontak yang terdapat pada perangkat pengguna, jadi dengan adanya kode *permission* ini semua isi dari kontak seperti nomor telepon dan email akan dapat dengan mudah untuk diakses [4].

Dalam mendeteksi anomali membutuhkan sebuah metode untuk membantu dalam pembuatan aplikasi deteksi anomali. Dalam penelitian ini metode yang digunakan yaitu menggunakan metode *prototyping*. Metode *prototyping* merupakan salah satu proses pendekatan dalam merancang sebuah sistem yang dimana hasilnya langsung diterapkan dalam sebuah model tanpa harus menunggu semua sistem terselesaikan agar nantinya dapat dievaluasi oleh pengguna dan hasil dari

evaluasi tersebut akan dijadikan acuan pada sistem yang akan dibuat [5].

Dalam pembuatan kode program biasanya kode yang dibuat sangat kompleks karena terdiri dari ribuan sampai jutaan baris kode namun tak jarang para *programmer* mendapatkan masalah pada kode program yang telah dibuat [6]. Salah satu penyebab dari masalah ini adalah karena adanya anomali. Anomali ini dapat membuat kode program bermasalah dari skala rendah hingga dampak yang parah pada sistem. Namun pada kenyataannya tidak semua *programmer* memiliki kemampuan untuk mendeteksi anomali secara cepat dan tepat sehingga membutuhkan sebuah aplikasi yang bisa untuk mendeteksi anomali pada kode program yang dibuat.

Perbedaan penelitian ini dengan penelitian sebelumnya adalah objek yang akan diteliti, pada penelitian sebelumnya itu hanya mendeteksi *error* pada kode program seperti *Syntax Error*, *Run-time Error*, dan *Logical Error*. Sedangkan pada penelitian ini mendeteksi anomali pada kode program yang dapat mengakibatkan permasalahan yang serius, seperti terkena virus, kerusakan pada sistem komputer, pencurian data atau informasi, dan masalah lain yang sewaktu-waktu dapat dilakukan oleh pihak yang tidak bertanggung jawab [7].

Dari permasalahan di atas, peneliti bertujuan untuk membuat aplikasi deteksi anomali untuk membantu para pengembang aplikasi dalam mendeteksi anomali pada kode program dengan judul “APLIKASI DETEKSI ANOMALI KODE PROGRAM MENGGUNAKAN METODE PROTOTYPING”. Aplikasi ini dibuat pada teks editor VSCode (*Visual Studio Code*) dalam penulisan kode programnya, lalu menggunakan bahasa pemrograman JavaScript untuk menambahkan interaksi agar lebih dinamis, HTML (*Hypertext Markup Language*) untuk membuat struktur dari kode program, CSS (*Cascading Style Sheets*), untuk memperindah tampilan pada aplikasi, dan PHP untuk menghubungkan ke database. Pada penelitian ini menggunakan metode *prototyping* dengan harapan dapat membantu para pengembang untuk lebih awal mengidentifikasi dan mengatasi potensi dari masalah tersebut, sehingga meningkatkan kualitas, dan keamanan perangkat lunak yang dihasilkan.

Oleh karena itu, penelitian ini bertujuan untuk membuat sebuah aplikasi deteksi anomali kode program menggunakan metode *prototyping*. Dengan menerapkan metode ini, diharapkan dapat ditemukan solusi yang efektif dan efisien dalam mengidentifikasi dan memperbaiki masalah yang ada dalam kode program secara cepat.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, ditetapkan rumusan masalah pada penelitian ini yaitu belum adanya aplikasi yang dapat mendeteksi anomali pada suatu kode program yang berbasis web.

1.3 Pertanyaan Penelitian

Berdasarkan rumusan masalah sebelumnya, maka diperoleh pertanyaan penelitian tentang “Bagaimana cara merancang sebuah aplikasi deteksi anomali kode program menggunakan metode *prototyping*?”

1.4 Batasan Masalah

Batasan masalah dari penelitian ini yaitu:

1. Penelitian berfokus pada deteksi dan *scanning* anomali yang terjadi dalam kode program.
2. Metode yang akan digunakan dalam penelitian ini yaitu menggunakan metode *prototyping* sebagai pendekatan dalam pengembangan aplikasi deteksi anomali kode program.
3. Bahasa pemrograman yang akan digunakan dalam pembuatan aplikasi ini yaitu JavaScript, HTML, dan PHP.
4. Bahasa pemrograman yang akan dideteksi meliputi bahasa yang digunakan pada aplikasi berbasis *mobile* seperti, Kotlin, C/C++, Java, Python.
5. Kode program anomali didapatkan dari referensi-referensi yang ada di internet.
6. Ekstensi file yang dapat dideteksi yaitu TXT dan XML.

7. Sampel anomali yang ditambahkan ke dalam database ada sepuluh sampel.

1.5 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mendeteksi anomali dan melakukan *scanning* pada sebuah kode program serta memperlihatkan contoh anomali yang mungkin saja bisa terjadi dalam kode program, sehingga *programmer* atau pengembang aplikasi dapat mengidentifikasi potensi masalah pada kode program sebelum diimplementasikan secara keseluruhan.

1.6 Manfaat Penelitian

Adapun manfaat dari penelitian ini berupa :

1. Dapat mengurangi kegagalan sistem dengan memperbaiki anomali sejak awal, dengan itu pengguna dapat menghindari masalah yang serius yang akan terjadi di masa depan.
2. Dapat meningkatkan kepuasan pengguna. Pengguna akan membuat perangkat lunak yang lebih aman melalui deteksi anomali kode program.