

## BAB 1 PENDAHULUAN

### 1.1. Latar Belakang

Risiko besar terhadap kegagalan rentan terjadi pada proses pengembangan perangkat lunak, mengingat adanya kompleksitas dari setiap tahapannya [1]. Untuk meminimalkan risiko tersebut, peran dari *Software Quality Assurance* (SQA) sangat krusial dalam memastikan kualitas perangkat lunak. Salah satu aspek penting dari SQA yakni proses pengujian perangkat lunak yang membutuhkan artefak yang disebut *test case*. *Test case* merupakan panduan spesifik yang menjadi acuan apakah perangkat lunak telah sesuai dengan spesifikasi yang diinginkan [2]. Proses pembuatan *test case* ini memerlukan waktu yang panjang, ketelitian, kreativitas, dan keahlian teknis terkait perangkat lunak yang diuji [3].

Berdasarkan dengan penelitian yang dilakukan oleh Dudekula dan teman-teman [4], menyatakan bahwa sekitar 58% praktisi tidak setuju akan automation tools meningkatkan deteksi kesalahan pada proses testing. Salah satu dari praktisi menunjukkan bahwa automation tools dapat digunakan tergantung bagaimana dengan cara penggunaannya [4]. Hal ini berkaitan dengan dalam praktik, pembuatan *test case* secara otomatis menggunakan *tools* mengalami beberapa kendala dalam menerjemahkan perintah yang diberikan [5].

Tools otomatisasi yang tersedia saat ini mengalami kesulitan dalam memahami *requirement* yang ada, terutama ketika perintah yang diberikan tidak menjelaskan secara detail [5]. Dengan adanya faktor kekurangan yang telah disebutkan sebelumnya, kebutuhan dalam peningkatan pemahaman *requirement* pada model dibutuhkan. Dengan adanya kemajuan teknologi, kecerdasan buatan (AI) menawarkan solusi untuk mengotomatisasi banyak proses yang sebelumnya dilakukan secara manual, termasuk pembuatan *test case* [6]. Berdasarkan penelitian yang dilakukan oleh Olsthoorn menyatakan bahwa *Automated Test Case Generation* dapat membantu pengembang perangkat lunak dengan mempercepat proses pembuatan *test case* [6].

*Automated Test Case Generation* bekerja dengan mengembangkan serangkaian *test case* yang dihasilkan secara acak untuk mengoptimalkan cakupan kode sesuai kriteria yang telah ditentukan [5]. Salah satu contoh

teknologi AI yang berpotensi besar dalam membantu proses ini adalah GPT. GPT merupakan *tools* AI yang telah digunakan dalam aktivitas sehari-hari seperti proses pembuatan teks, penerjemah bahasa dan *automated customer support* [7]. Dimana GPT telah banyak diteliti sebagai *tools* yang dapat membantu menghasilkan *test case*. Berdasarkan dengan penelitian yang dilakukan oleh Gaolei Yi dan teman-teman *test case* yang dihasilkan oleh GPT memiliki tingkat keterbacaan yang tinggi namun *test case* yang dihasilkan menunjukkan terjadinya keacakan atau adanya variasi [8]. Variasi ini menunjukkan bahwa kurangnya pemahaman model dalam memahami *requirement* yang diberikan, sehingga mempengaruhi konsistensi dari *test case* yang dihasilkan GPT. Oleh karena itu, diperlukan penyesuaian lebih lanjut, dengan memberikan pemahaman tambahan mengenai *requirement* pada model GPT melalui *continual pretraining* [9]. *Continual pretraining* adalah proses pelatihan lanjutan pada model bahasa besar yang dilakukan secara bertahap dan berkelanjutan, dengan tujuan memperbarui dan memperluas pengetahuan model terhadap informasi atau domain baru, tanpa harus melatih ulang model dari awal [10]. Hal ini digunakan untuk mengatasi tantangan terutama dalam meningkatkan pemahaman model dari ChatGPT dalam memahami *requirement* untuk menghasilkan *test case*.

Selain dengan memberikan pemahaman baru pada model untuk meningkatkan pemahaman *requirement* model ChatGPT dalam menghasilkan *test case*, ChatGPT dapat dikombinasikan dengan *Natural Language Process* (NLP) yang berpotensi mengatasi adanya ambiguitas pada *requirement* perangkat lunak [11]. *Natural Language Process* (NLP) merupakan bidang yang membantu komputer dalam memahami teks dan ucapan suara [12]. NLP dapat digunakan sebagai perantara untuk menganalisis dokumen dari kebutuhan dan mengekstraksi kebutuhan data untuk membuat *test case*. Hal ini dapat meningkatkan kualitas *test case* yang dihasilkan lebih tinggi dan konsisten sehingga meningkatkan efektivitas pembuatan *test case* [13]. Dengan adanya ketidak konsistenan *test case* yang dihasilkan oleh ChatGPT, maka penelitian ini bertujuan untuk meningkatkan pemahaman model terhadap *requirement* agar meningkatkan konsistensi dan *requirement coverage* dari GPT dalam

menghasilkan *test case* serta membandingkan hasil dari *test case* yang dihasilkan oleh GPT yang telah melalui pemahaman ulang *requirement* melalui proses pelatihan ulang dan dikombinasikan oleh NLP dengan hasil *test case* yang dihasilkan oleh model dasar ChatGPT.

## **1.2. Rumusan Masalah**

- 1.2.1. Bagaimana kemampuan model dasar GPT dalam memahami *requirement* perangkat lunak dan menghasilkan *test case* yang sesuai?
- 1.2.2. Bagaimana pengaruh penambahan pengetahuan *requirement* pada model GPT terhadap konsistensi *test case* yang dihasilkan ?

## **1.3. Tujuan Penelitian**

- 1.3.1. Mengevaluasi kemampuan model dasar GPT dalam memahami *requirement* perangkat lunak dan menghasilkan *test case* yang relevan untuk otomatisasi pengujian.
- 1.3.2. Menganalisis sejauh mana penambahan pengetahuan *requirement* dan teknik NLP dapat meningkatkan konsistensi *test case* yang dihasilkan pada model GPT.

## **1.4. Manfaat Penelitian**

- 1.4.1. Memberikan pemahaman mengenai penambahan pengetahuan dalam meningkatkan kemampuan model GPT memahami dan menerjemahkan *requirement* ke dalam *test case*.
- 1.4.2. Menyediakan pendekatan berbasis LLM yang dapat digunakan untuk menghasilkan *test case* secara otomatis dengan cakupan *requirement* yang tinggi dan struktur yang konsisten.

## **1.5. Rencana Kegiatan**

Rencana kegiatan pada penelitian ini adalah melakukan studi literatur, analisis dan perancangan data, perancangan model, perancangan kombinasi teknik kecerdasan buatan, analisis dan evaluasi model awal, dan penulisan laporan TA .

## **1.6. Jadwal Kegiatan**

Pelaksanaan kegiatan dalam Tugas Akhir ini direncanakan berlangsung selama kurang lebih 6 bulan mengikuti jadwal semester. Kegiatan disusun secara sistematis guna memastikan seluruh tahapan dapat diselesaikan tepat waktu sesuai dengan rencana yang telah ditetapkan.

Adapun jadwal kegiatan secara rinci disajikan pada Tabel 1.1 berikut:

**Tabel 1.1 Jadwal Kegiatan Penelitian**

Kegiatan	Bulan				
	1	2	3	4	5
Studi literatur					
Analisis dan perancangan data					
Perancangan model					
Perancangan kombinasi teknis kecerdasan buatan					
Analisis dan evaluasi model awal					
Analisis Perbandingan hasil evaluasi					
Penulisan Laporan TA					