

BAB 1 PENDAHULUAN

Bagian pendahuluan memuat beberapa substansi sebagai berikut:

1.1. Latar Belakang

Dalam pengembangan perangkat lunak modern, proses memperbaiki program (*program repair*) masih menjadi suatu aktivitas yang secara besar dilakukan secara manual, sehingga memakan banyak waktu dan biaya [1], [2]. Salah satu topik penelitian di bidang perangkat lunak yang relevan untuk mengatasi tantangan ini adalah Automated Program Repair (APR), yang bertujuan umum untuk mengurangi upaya manual dalam memperbaiki *bug* perangkat lunak [3], [4], [5]. Dengan mengotomatiskan proses perbaikan, APR tidak hanya membantu mengurangi waktu yang diperlukan, tetapi juga meminimalkan potensi kesalahan akibat intervensi manual. Di sisi lain, Large Language Models (LLMs) telah muncul sebagai terobosan inovatif dalam mendukung pengembangan perangkat lunak. Dalam lingkungan penggunaan LLM, pengembang dapat berkomunikasi dengan sistem melalui dialog, di mana mereka mengajukan pertanyaan atau mendeskripsikan hasil yang diinginkan tanpa perlu menjelaskan secara eksplisit cara mencapainya [6], [7]. Pendekatan ini memungkinkan pengembang untuk berinteraksi dengan sistem secara lebih alami, membuka peluang baru dalam perbaikan program dengan cara yang lebih intuitif dan efisien.

LLM dapat mengatasi berbagai kendala klasik APR, seperti keterbatasan variasi *patch*. Kendala ini muncul karena ketergantungan pada *bug-fixing dataset* untuk APR tradisional atau prediksi *patch* langsung untuk *learning-based APR* [4]. LLM mampu memanfaatkan *dataset* yang luas untuk memahami, menganalisis, dan menghasilkan kode dengan komprehensif. Kemampuan ini memungkinkan LLM untuk memberikan berbagai saran kepada pengembang perangkat lunak untuk memperbaiki *bug*. Oleh karena itu, penggunaan LLM menandai kemunculan paradigma baru dalam penelitian APR, di mana teknologi ini secara signifikan meningkatkan

efektivitas dan fleksibilitas dalam proses perbaikan program. Efektivitas perbaikan ini ditunjukkan melalui penelitian Xia dkk. [8]. yang berhasil menghasilkan 3.3x lebih banyak perbaikan dibandingkan *baseline* yang ditetapkan oleh *database bugs* Defects4J (Versi 2.0) [9].

Saat ini, berbagai penelitian telah muncul untuk membahas mengenai penerapan sistem APR yang memanfaatkan LLM untuk membantu proses perbaikan *bug* [4], [10]. dan juga menjelaskan bahwa salah satu potensi *future work* adalah penggunaan *dialogue-based* LLM untuk APR seperti ChatGPT. Namun, potensi ini menghadapi hambatan utama berupa kurangnya suatu alur yang jelas untuk memandu pengembang dalam implementasi potensi tersebut di konteks pengembangan perangkat lunak nyata untuk alur kerja/*workflow* yang terdistribusi. Selain itu, LLM yang merupakan bagian dari Machine Learning (ML) memiliki masalah tersendiri yang perlu diperhatikan seperti kurangnya penerimaan pengguna hingga halusinasi model [11], [12].

Oleh karena itu, penelitian ini berfokus pada perancangan suatu alur penerapan APR berbasis LLM yang didasarkan pada *design pattern* yang jelas dan kompatibel dengan alur pengembangan terdistribusi modern agar dapat diimplementasikan. Alur penerapan APR berbasis LLM beserta pola desain yang dirancang akan menggunakan pendekatan/*design pattern* Human-in-the-Loop (HiL) [11] di mana seorang pengembang perangkat lunak dapat melibatkan dirinya secara langsung ke dalam *workflow* penerapan APR berbasis LLM tersebut. Penggunaan pendekatan ini akan mengatasi/memitigasi berbagai masalah dari ML mulai dari kurangnya penerimaan pengguna hingga halusinasi model karena adanya pemanfaatan kemampuan umpan balik (*feedback loop*) LLM dengan manusia. Pendekatan ini juga akan selaras dengan adanya suatu kebutuhan *framework* yang solid untuk kolaborasi manusia dan AI/LLM yang berguna untuk mengarahkan dan memvalidasi keluaran AI/LLM tersebut [13].

1.2. Rumusan Masalah

Permasalahan utama dalam implementasi APR berbasis LLM adalah kurangnya suatu panduan berbentuk alur yang didasarkan pada HiL *design pattern* yang jelas untuk memudahkan pengembang perangkat lunak dalam mengintegrasikan teknologi ini secara efektif ke dalam alur pengembangan nyata yang terdistribusi. Selain itu, LLM sebagai bagian dari LLM memiliki masalah tersendiri seperti kurangnya penerimaan pengguna hingga halusinasi model sehingga menuntut adanya pendekatan yang melibatkan manusia dalam interaksi LLM (seperti pendekatan Human-in-the-Loop) untuk memitigasi batasan tersebut. Adapun rumusan masalah dari penelitian ini dibagi menjadi beberapa pertanyaan penelitian (PP) adalah sebagai berikut:

- PP1: Bagaimana cara pengembangan alur untuk penerapan Human-in-the-Loop APR berbasis LLM yang efektif dan terstruktur?
- PP2: Bagaimana sistem hasil penerapan Human-in-the-Loop APR berbasis LLM bekerja terkait hal efisiensi dan penerimaan pengembang?

1.3. Tujuan dan Manfaat

Tujuan penelitian ini adalah untuk menghasilkan suatu alur penerapan Human-in-the-Loop APR berbasis LLM yang terintegrasi dengan HiL *design pattern* yang jelas, kompatibel dengan arsitektur pengembangan terdistribusi, serta tervalidasi dalam suatu model sistem. Hasil penelitian ini diharapkan dapat digunakan sebagai panduan untuk para pengembang perangkat lunak yang ingin menerapkan sistem APR berbasis LLM untuk mempermudah kegiatan perbaikan perangkat lunak.

1.4. Batasan Masalah

Adapun batasan dari penelitian ini adalah sebagai berikut:

1. Sebagai *proof of concept*, penerapan alur Human-in-the-Loop APR berbasis LLM, penelitian menggunakan GitHub Actions sebagai lingkungan pengembangan terdistribusi penerapan alur tersebut.

2. Penelitian menggunakan *dataset* Refactory [14] dan QuixBugs[15]. yang berbasis bahasa pemrograman Python untuk evaluasi sistem Human-in-the-Loop APR berbasis LLM. Meskipun demikian, sistem ini dapat mendukung *dataset* dengan bahasa pemrograman lainnya.

1.5. Metode Penelitian

Metode penelitian untuk menyelesaikan pekerjaan di dalam Tugas Akhir adalah dengan menjalankan suatu alur penelitian yang telah didefinisikan dengan penjelasan singkat sebagai berikut:

1. Analisis *design pattern* untuk LLM: Merupakan tahapan awal yang dilakukan untuk mengidentifikasi *design pattern* pada LLM yang sesuai dengan alur sistem APR yang akan dikembangkan sehingga dapat mendukung interaksi manusia dengan LLM dalam proses sistem APR.
2. Perancangan alur Human-in-the-Loop APR berbasis LLM: Tahap ini merupakan bagian pengembangan alur sistem Human-in-the-Loop APR berbasis LLM yang didasari oleh hasil analisis pada tahapan sebelumnya dengan batasan penelitian yang telah ditetapkan.
3. Implementasi sistem Human-in-the-Loop APR berbasis LLM: Alur yang telah dikembangkan sebelumnya akan digunakan pada tahapan ini untuk diimplementasikan menjadi suatu program pada lingkungan pengembangan terdistribusi, yaitu GitHub Actions.
4. Evaluasi hasil implementasi Human-in-the-Loop APR berbasis LLM: Pada tahapan akhir ini, program yang telah dikembangkan akan dievaluasi efektivitasnya menggunakan metode eksperimen dan wawancara.
5. Penulisan laporan mengenai kegiatan penelitian yang telah dilakukan.