

## 1. Pendahuluan

### Latar Belakang

*Requirements traceability* adalah salah satu tugas yang paling penting dan menantang di industri untuk memastikan bahwa perangkat lunak telah dikembangkan sesuai dengan semua kebutuhan pengguna[1]. Kebutuhan perlu ditentukan dan ditelusuri sepanjang siklus hidup pengembangan perangkat lunak untuk memetakan *link* antara artefak perangkat lunak yang berbeda. Selain itu, *traceability links* menyediakan metode untuk *requirements* perangkat lunak agar sesuai dengan spesifikasi, kontrak, atau peraturan. Di tingkat organisasi, mengelola *links* tersebut bertindak sebagai investasi untuk meningkatkan kualitas produk, mengurangi biaya pemeliharaan, dan memfasilitasi penggunaan kembali[2].

Sayangnya, metode manual *traceability* tidak cocok untuk kebutuhan industri rekayasa perangkat lunak[3]. Jumlah *link traceability* yang perlu ditangkap akan tumbuh secara eksponensial, sesuai dengan ukuran dan kompleksitas sistem perangkat lunak[4]. Ini berarti mendeteksi *traceability* secara manual untuk proyek perangkat lunak besar membutuhkan banyak waktu dan upaya. Metode manual *traceability* juga rentan terhadap perubahan sistem. Jika terjadi perubahan pada elemen apa pun yang ditangkap dalam data *traceability*, bagian yang terpengaruh dari data *traceability* harus diperbarui secara manual. Hal ini membutuhkan kedisiplinan dan sejumlah waktu dan upaya yang besar yang dihabiskan untuk pembaharuan *link* di seluruh data *traceability*. Karena itu, mudah untuk membuat data *traceability* secara manual menjadi tidak selaras dengan kumpulan *requirements* dan *test cases* saat ini. Di sisi lain, *tracing requirement* secara manual tidak sepenuhnya menguntungkan *development* perangkat lunak[5].

Untuk mengatasi masalah-masalah tersebut, otomatisasi proses *tracing* memungkinkan proses *development* lebih cepat dan praktis untuk mendapatkan keuntungan dari *requirements tracing*. Berdasarkan penelitian, salah satu metode yang dapat digunakan dalam melakukan *requirements tracing* adalah dengan menggunakan metode *information retrieval* (IR). Metode IR yang digunakan adalah metode *Tf-Idf*. Metode ini digunakan karena metode ini merupakan metode pembobotan kata yang dikenal cukup baik, mudah dan memiliki hasil yang akurat[6]. Dengan mencari *accuracy*, *precision* dan *recall* akan performansi dari hasil yang dicari dalam penelitian ini. *Accuracy* sendiri adalah derajat kedekatan atau semakin dekat tembakan ke sasaran, semakin akurat[7]. Kemudian, untuk *precision* adalah pengulangan atau semakin banyak jumlah bidikan yang bisa didapatkan yang saling berdekatan maka semakin presisi hasil yang didapat. Terakhir, *recall* adalah Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

Namun sayangnya pada penelitian terdahulu tidak ditunjukkan performansi dari hasil penerapan metode *Tf-Idf* tersebut. Oleh karena itu, pada penelitian ini dilakukan penelitian mengenai performa metode *Tf-Idf* dalam mengembalikan ketelusuran antara *requirements* dan *test cases*.

### Rumusan Masalah

Pada penelitian ini didefinisikan pertanyaan-pertanyaan penelitian (PP) sebagai berikut:

- PP1. Seberapa besar tingkat *accuracy* yang didapatkan ketika menggunakan metode *Tf-Idf*?
- PP2. Seberapa besar tingkat *precision* yang didapatkan ketika menggunakan metode *Tf-Idf*?
- PP3. Seberapa besar tingkat *recall* yang didapatkan ketika menggunakan metode *Tf-Idf*?

### Tujuan

Penelitian ini bertujuan untuk menganalisis, menghitung dan membandingkan apakah *Tf-Idf* dapat membantu mengembalikan *requirement traceability*. Penelitian ini memberikan kontribusi terhadap Badan Ilmu Rekayasa Perangkat Lunak terutama pada area pengujian dan kebutuhan perangkat lunak dalam bentuk pengetahuan akan metode yang dapat digunakan untuk menelusuri keterkaitan kebutuhan dengan *test cases*.

### Topik dan Batasannya

Metode yang digunakan merupakan TF-IDF (Term Frequency Inverse Document Frequency) dan dataset yang digunakan adalah *requirement*, *test cases* dan *golden standard* EBT dari CoEST[8]. Tanpa mengubah dataset, Bahasa yang digunakan pada dataset *requirement*, *test cases* dan *golden standard* adalah Bahasa Inggris.

### Organisasi Tulisan

Setelah Pendahuluan, dilanjutkan dengan bagian ke-2 Studi Terkait yang berisi teori/studi/literatur yang mendukung (terkait erat) dengan penelitian yang dikerjakan. Pada bagian ke-3 Sistem yang Dibangun yaitu menjelaskan rancangan dan sistem atau produk yang dihasilkan. Dilanjutkan bagian ke-4 Evaluasi yaitu Hasil Pengujian dan Analisis Hasil Pengujian. Dan diakhiri pada Bagian ke-5 Kesimpulan yang memuat kesimpulan dan Saran (*Future Work*)