

1. Pendahuluan

Latar Belakang

Tracing Requirements ke dalam *source code* merupakan aspek penting dari rekayasa perangkat lunak. *Traceability* memberikan informasi yang bermanfaat untuk menganalisis setiap biaya dan manfaat *requirements* yang diterapkan baik dalam tahap pengembangan dan pemeliharaan. 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[1, 2]. Namun, mengelola secara manual *requirements traceability* tersebut dapat menjadi tantangan.

Mengelola *traceability* secara manual sulit dicapai untuk beberapa tantangan. *Traceability* proyek cenderung menurun selama evolusi dan pemeliharaan perangkat lunak sebagai perubahan *source code*[3]. Artefak perangkat lunak, termasuk *requirements specification* perangkat lunak, cenderung ketinggalan zaman antara interval pendek *maintenance* dan evolusi, yang mengarah ke *links* yang salah antara *requirements* lama dan *source code* yang baru diterapkan. Selain itu, identifikasi manual dari *traceability links* rentan terhadap kesalahan manusia[4]. Kondisi ini menyebabkan diperlukannya otomatisasi dalam mengidentifikasi *traceability links*.

Salah satu cara untuk mengotomatiskan *traceability links* penelusuran adalah dengan menerapkan metode pengambilan informasi, seperti istilah *term frequency-inverse document frequency* (TF-IDF), antara *requirements* perangkat lunak dan *source code*. TF-IDF adalah statistik numerik yang mencerminkan betapa pentingnya sebuah kata bagi sebuah dokumen dalam suatu koleksi. TF-IDF adalah salah satu metode pencarian informasi yang menonjol karena sederhana dan efektif[5].

Topik dan Batasannya

Penelitian ini menitikberatkan pada implementasi metode TF-IDF untuk identifikasi *traceability link* antara *requirement* dengan *source code* pada perangkat lunak. Hasil dari perhitungan TF-IDF terhadap beberapa kasus dan dataset selanjutnya akan dianalisis, sebagai dasar melakukan modifikasi terhadap tahapan dalam *requirements traceability* antara *requirement* dengan *source code* pada perangkat lunak. Untuk mencapai tujuan tersebut, penulis menerapkan metode dan mengujinya menggunakan kumpulan data. Untuk evaluasi metode TF-IDF.

Tujuan

- Tujuan yang tertera digunakan untuk memandu penulis dalam penelitian, yang dimana tujuannya seperti berikut:
- Bagaimana cara melakukan recovery antara requirement dengan source code dengan menggunakan TF-IDF?
 - Mengukur accuracy, precision, dan recall dari metode TF-IDF yang digunakan.

Penelitian penulis berkontribusi pada badan pengetahuan rekayasa perangkat lunak, terutama bidang pengetahuan teknik *requirements*, untuk memahami cara mengelola *requirement traceability* dengan lebih efisien.

Organisasi Tulisan

Sisa dari makalah ini dibangun sebagai berikut: Bab 2 menjelaskan pengetahuan kita tentang teori latar belakang dan pekerjaan terkait penelitian ini. Bab 3 menjelaskan penelitian yang telah penulis lakukan. Bab 4 menyajikan hasil penelitian penulis dan membahasnya. Akhirnya, penulis menyimpulkan penelitian ini di Bab 5 dengan beberapa karya mendatang.