

1. Pendahuluan

1.1 Latar Belakang

Use Case Description ditulis menggunakan *Natural Language (NL)* yang merupakan bahasa inti dari mesin penerjemah. *step performed* pada *use case description* membantu memahami variabilitas dari sudut pandang *use case*. Serta mendeskripsikan aktivitas yang terjadi dalam *use case* berupa skenario [1]. *Use case description* merupakan sebuah aktivitas yang melibatkan proses pengembangan sebuah perangkat lunak. Pada penelitian ini terdapat beberapa komponen pada *use case description*, yaitu: *Use case name, Actor, Step Performed, Pre-Condition, Post Condition, dan Assumption* [2].

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah sebuah algoritma yang digunakan dalam mengekstraksi teks atau sebuah teknik yang digunakan untuk memberikan bobot yang tepat antara sebuah kata (*term*) dan dokumen yang bersangkutan. Metode TF-IDF ini menggabungkan dua konsep penting dalam menghitung bobot tersebut, yaitu frekuensi kemunculan kata dalam dokumen serta frekuensi dokumen yang mengandung kata tersebut [3,4].

Proses perbandingan artefak dilakukan melalui pencarian similarity teks yang secara umum merupakan sebuah metode untuk mengukur kesesuaian atau kesamaan antara dua blok. Selain itu, *Elicitation* adalah sebuah proses yang bertujuan untuk mendefinisikan *requirement*, membuat keputusan yang sesuai dengan klien. Pengembang dan klien berdiskusi untuk menghasilkan *requirement* yang sesuai untuk memenuhi tujuan dan menghindari kegagalan diakhir pengembangan perangkat lunak. Elisitasi memiliki beberapa metode, seperti: *Interviews, Prototype, Brainstorming, Questionnaire, Social Network, Feedback, dan Comment Analysis* dan lain sebagainya. Metode elisitasi pada penelitian ini menggunakan metode *interviews* untuk mendapatkan data kebutuhan dari klien serta menganalisis kebutuhannya [5, 6].

Aktivitas perbandingan yang digunakan pada penelitian ini menggunakan Ekstraksi melalui *Text Pre-Processing*, yang berguna untuk mempersiapkan data teks dalam mengurangi noise pada data. Pada penelitian ini, *Text Pre-Processing* digunakan untuk memproses dan menganalisis *text* yang sudah dimodelkan serta menerapkan metode *Text Mining*. [7]

SRS (*Software Requirement Specification*) adalah sebuah detail requirement yang mendefinisikan, mengumpulkan, dan menyajikan informasi untuk membuat sebuah perangkat lunak. SRS juga memberi kemudahan untuk mengembangkan sebuah perangkat lunak yang menggambarkan kebutuhan sebelum pengembangan perangkat lunak dimulai. Fungsi penggunaan bahasa alami pada SRS untuk memudahkan dalam memahami pendefinisian kendala dari sebuah dokumen yang terdiri dari deskripsi menyeluruh tentang sistem yang akan dikembangkan. SRS juga memiliki kesalahan atau cacat pada sistem yang dimana harus segera terdeteksi agar tidak mempengaruhi kualitas dokumentasi serta menambah biaya dan menghemat waktu [8, 9].

1.2 Topik dan Batasannya

Topik dan Batasan pada tugas akhir, sebagai berikut:

1. Bagaimana cara mengimplementasikan *Text Pre-Processing* pada hasil ekstraksi *Step Performed*.
2. Bagaimana proses dalam mendapatkan hasil *validity*.
3. Dataset pada penelitian ini menggunakan Bahasa Inggris.

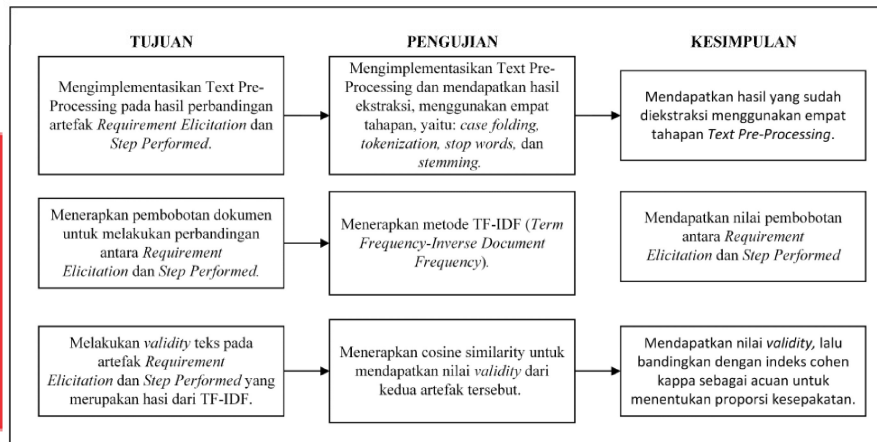
1.3 Tujuan

Tugas akhir ini secara garis besar memiliki tujuan untuk mengekstraksi kedua artefak, sebagai berikut:

1. Mengimplementasikan *Text Pre-Processing* pada hasil perbandingan artefak *Requirement Elicitation* dan *Step Performed*.
2. Menerapkan pembobotan dokumen untuk melakukan perbandingan antara *Requirement Elicitation* dan *Step Performed*.
3. Melakukan *validity* teks pada artefak *Requirement Elicitation* dan *Step Performed* yang merupakan hasil TF-IDF.

1.4 Organisasi Tulisan

Pada sub-bagian ini, terdapat beberapa hal yang dibahas. Dapat kita lihat pada pendahuluan memperkenalkan konsep *use case description*, beserta komponen-komponennya, serta membahas TF-IDF dalam ekstraksi teks. Bagian selanjutnya, proses perbandingan artefak yang membahas metode pengukuran kesesuaian teks, serta memperlihatkan pentingnya elisitasi dan metode *interview* dalam pengumpulan data kebutuhan. Selanjutnya, aktivitas perbandingan memberikan contoh penggunaan ekstraksi melalui *Text Pre-Processing* dan penerapan *Text Mining*. Pada SRS (*Software Requirement Specification*) memiliki fokus pada detail *Requirement Elicitation* dan peran SRS dalam pengembangan perangkat lunak dan menekankan pentingnya mendeteksi kesalahan pada sistem untuk memastikan kualitas dokumen.



Gambar 1 Keterkaitan antara tujuan, pengujian, dan kesimpulan

