

2. Studi Terkait

2.1. Penelitian Terkait

Klasifikasi teks merupakan salah satu tugas *Natural Language Processing* (NLP) yang melibatkan pengelompokkan teks berdasarkan isinya ke dalam kategori-kategori yang telah ditentukan sebelumnya. Tugas ini telah berkembang menjadi tugas-tugas spesifik lainnya, termasuk analisis sentiment [19], penyaringan spam [20] dan pemodelan topik [21].

Penelitian klasifikasi pada kebutuhan perangkat lunak juga telah aktif dilakukan. Penelitian tersebut umumnya digunakan untuk mengklasifikasikan aspek-aspek kebutuhan berdasarkan tipe dan karakteristiknya. Model klasifikasi ini digunakan untuk mengategorikan kebutuhan sebagai fungsional atau non-fungsional [12], [13], [22] dan untuk lebih lanjut menentukan karakteristiknya, seperti atomisitas [14], konsistensi [15], atau yang berkaitan dengan keamanan [16].

Mengmeng dkk. [12] mengusulkan model klasifikasi untuk kebutuhan yang membaginya menjadi tiga kategori: non-fungsional (misalnya, keandalan, kegunaan, portabilitas, dan kinerja), fungsional, dan lain-lain. Penelitian ini menggunakan pendekatan Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Chi-Squared (CHI), dan AUR-BoW dengan tiga model pembelajaran mesin: Naive Bayes, J28, dan Bagging. Penelitian ini mencapai hasil terbaik pada presisi (71,4%), recall (72,3%), dan F-measure (71,8%).

Sebaliknya, Rahman dkk. [13] fokus secara eksklusif pada mengklasifikasikan kebutuhan non-fungsional ke dalam sebelas kategori: ketersediaan, legalitas, tampilan dan nuansa, keterjagaan, operasional, kinerja, skalabilitas, keamanan, kegunaan, toleransi kesalahan, dan portabilitas. Penelitian ini menggunakan pendekatan klasifikasi multi-kelas dengan dua varian model Recurrent Neural Network (RNN) - Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU). Menurut temuan, model LSTM memiliki presisi tertinggi (97,3%), recall (96,7%), dan skor F1 (96,6%).

Penelitian saat ini berfokus pada kasus-kasus kebutuhan yang mencakup klasifikasi fungsional/non-fungsional, atomisitas, konsistensi, atau aspek yang berkaitan dengan keamanan. Namun, penelitian ini tidak fokus pada mengklasifikasikan kasus uji di dalam kebutuhan. Oleh karena itu, penelitian ini bertujuan untuk mengisi kesenjangan ini dengan mengeksplorasi area tersebut dan memberikan wawasan mengenai klasifikasi kasus uji dalam kebutuhan.

2.2. Kebutuhan perangkat Lunak

Kebutuhan perangkat lunak adalah kemampuan perangkat lunak yang dibutuhkan oleh pengguna untuk menyelesaikan masalah atau mencapai objektif. Kemampuan tersebut harus dapat dipenuhi atau dimiliki oleh sebuah sistem, komponen, atau layanan untuk memenuhi standar, perjanjian atau dokumen lainnya [2]. Wiegers [23] membagi kebutuhan perangkat lunak menjadi 3 tingkat: *business requirement*, *user requirement*, dan *functional requirement*. Tingkat tersebut dideskripsikan sebagai berikut.

- *Business requirement*: Tujuan bisnis tingkat tinggi yang berasal dari organisasi yang membangun produk atau pelanggan yang membeli produk tersebut.
- *User requirement*: Tujuan atau tugas yang harus bisa dilakukan oleh kelas spesifik pengguna atau atribut produk yang diinginkan pada sistem.
- *Functional requirement*: Deskripsi perilaku yang akan ditunjukkan oleh sistem dalam kondisi tertentu.

2.3. Pola Penulisan dalam Kebutuhan Perangkat Lunak Berbahasa Alami

Kebutuhan perangkat lunak terdiri atas spesifikasi formal dan informal. Spesifikasi formal dapat dituliskan menggunakan *history-based*, *state-based*, *transition-based*, *functional*, atau *operational specification* [24]. Sedangkan spesifikasi informal atau bahasa alami dapat menggunakan perspektif pengguna, atau sistem. Saat ini, beberapa pola penulisan kebutuhan berbahasa alami dapat digunakan seperti pola template Alexander dan Steven [3], EARS [25], dan *user stories* [26].

Pada penulisan kebutuhan berbahasa alami dengan perspektif pengguna, pola dapat menggunakan templat Alexander dan Steven [3]: “*The [user class or actor name] shall be able to [do something] [to some object] [qualifying conditions, response time, or quality statement]*”, atau menggunakan templat *user stories* [26]: “*As [aspect of who], I want/want to/need/can/would like [aspect of what], so that [aspect of why]*”. Sedangkan perspektif sistem dapat menggunakan templat EARS [25]: “*[optional precondition] [optional trigger event] the system shall [expected system response]*”.

Tabel 2. Contoh kalimat kebutuhan perangkat lunak berbahasa alami dalam template Alexander dan Steven [3].

<i>The viewer shall be able to Search videos by its category in under 30 seconds</i>					
<i>The viewer</i>	<i>shall be able to</i>	<i>Search</i>	<i>videos</i>	<i>by its category</i>	<i>in under 30 seconds</i>
<i>User class</i>	<i>Conjunction</i>	<i>Do something</i>	<i>To some</i>	<i>Qualifying conditions</i>	<i>Response time/Quality</i>