

			<i>Object</i>		<i>statement</i>
--	--	--	---------------	--	------------------

Tabel 3. Contoh kalimat kebutuhan perangkat lunak berbahasa alami dalam template *user stories* [26].

<i>As a Data Publishing User I want to be able to edit a dataset I have published so that I can correct or enhance existing data</i>		
<i>As a Data Publishing User</i>	<i>I want to be able to edit a dataset I have published</i>	<i>so that I can correct or enhance existing data</i>
<i>Aspect of who</i>	<i>Aspect of what</i>	<i>Aspect of why</i>

Tabel 4. Contoh kalimat kebutuhan perangkat lunak berbahasa alami dalam template EARS [25].

<i>In registration menu if the submit button is pressed the system shall validate the correctness of input</i>			
<i>In registration menu</i>	<i>if the submit button is pressed</i>	<i>the system shall</i>	<i>validate the correctness of input</i>
<i>Optional precondition</i>	<i>Optional trigger event</i>	<i>Conjunction</i>	<i>Expected system response</i>

2.4. Naïve bayes Classification

Naive Bayes adalah model statistik untuk klasifikasi teks yang banyak digunakan [27]. Model ini bergantung pada teorema Bayes sebagai prinsip panduan. Menurut aturan ini, model ini mendapatkan kekuatan melalui analisis data dan pengamatan, terutama dalam sejauh mana mereka secara efektif didukung oleh bukti yang tersedia. Oleh karena itu, pemodelan yang baik memerlukan analisis menyeluruh terhadap semua sumber data penting.

Klasifikasi *Naive Bayes* sederhana dan efektif dalam banyak situasi dunia nyata. Namun, salah satu asumsi utama dari model *Naive Bayes* adalah bahwa semua atribut yang digunakan untuk menggambarkan sebuah contoh adalah kondisional independen jika diberikan kelas dari contoh tersebut [28]. Dalam rekayasa perangkat lunak, *Naive Bayes* telah digunakan untuk berbagai tujuan seperti prediksi kesalahan perangkat lunak [29], penerbitan data yang menjaga privasi [30], dan prediksi jurusan mahasiswa [31]. *Naive Bayes* juga telah diintegrasikan ke dalam pengklasifikasi komposit untuk prediksi yang akurat [32].

2.4. Entitas Kasus Uji

Kasus uji tersusun atas komponen-komponen penting, diantaranya adalah langkah-langkah pengujian, masukan atau data uji, kondisi sebelum dan sesudah [8], [9]. Komponen tersebut dapat distandarkan menjadi sebuah tabel ataupun bentuk lain. Bentuk lain dari kasus uji dapat dibuat berdasarkan pendekatan berbasis perilaku pengguna, atau dikenal sebagai *Behavior-Driven Development* (BDD). BDD memiliki tujuan untuk kasus uji yang dapat dieksekusi dari kebutuhan sistem yang dapat dipahami oleh *stakeholder* [33]. Komponen umum pembentuk kasus uji BDD terdiri atas aktor dan kondisi pengujian. Kondisi pengujian tersebut terdiri atas: “*Given*”: kondisi sebelum pengujian atau keadaan awal, “*When*”: Aksi yang dilakukan, “*Then*”: kondisi sesudah pengujian atau hasil yang dicapai.

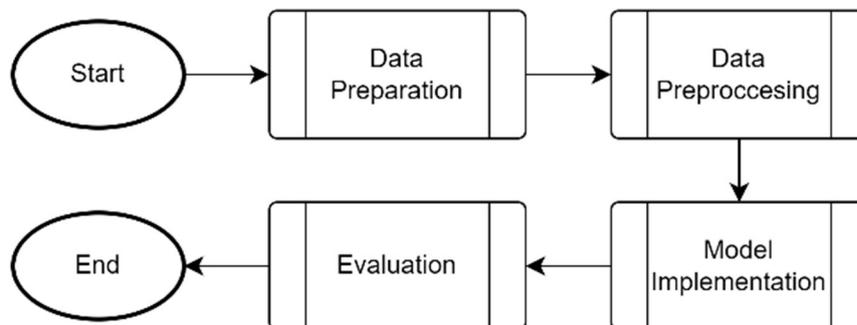
Penelitian ini menggunakan komponen BDD tersebut untuk mengkategorikan entitas pada dataset. Kategori utama didefinisikan sebagai *precondition* (*Given*), *action* (*When*), *postcondition* (*Then*), *actor* (*User/System*). Selain itu, entitas *condition* dan *quality* ditambahkan sebagai parameter pengukuran dalam pengujian. Kategori entitas didefinisikan pada tabel sebagai berikut.

Tabel 5 Entitas kasus uji

Entitas	Deskripsi	Contoh
Prakondisi	Kondisi sebelum pengujian atau keadaan awal	<i>In registration menu, While user on the landing page, When user at login menu</i>
Pascakondisi	Kondisi sesudah pengujian atau hasil yang dicapai	<i>the system shall validate the correctness of input, the system shall display “insert card to begin”</i>

3. Sistem yang Dibangun

Pada bagian ini, metode pendekatan yang digunakan untuk merancang model sistem ekstraksi entitas otomatis diusulkan. Sistem yang dibuat ditujukan untuk mengidentifikasi entitas kasus uji pada kebutuhan perangkat lunak berbahasa alami. Sistem dibangun dengan empat tahapan inti yaitu *dataset preparation*, *dataset preprocessing*, *model implementation*, dan *evaluasi*. Gambar 1 merupakan alur dari metode penelitian ini.



Gambar 1 Alur metode penelitian ekstraksi entitas kebutuhan perangkat lunak berbahasa alami

3.1. Dataset Preparation

Dataset yang digunakan dalam penelitian ini menggabungkan data dari dataset PURE [17] dan PROMISE [18] yang terdiri dari 2600 kebutuhan perangkat lunak berbahasa alami (Gambar 2). Dataset ini telah melalui proses *dataset preparation* yang terdiri dari tiga langkah: *data cleaning*, *manual annotation*, dan *data splitting*. Gambar 3 adalah alur proses dataset preparation.

- | |
|--|
| <ol style="list-style-type: none"> 1 When the ETCS trainborne equipment is isolated, the system shall not show any ETCS information other than the fact that the system is isolated. 2 Isolation of the ETCS trainborne equipment shall disconnect the ETCS trainborne equipment from the vehicle braking system. 3 The ETCS on-board shall be capable of receiving track description from the trackside. 4 The ETCS trainborne equipment shall supervise the end of movement authority, if this information is available on-board 5 If the release speed is calculated on board it shall ensure that the train will stop before reaching the danger point 6 The ETCS trainborne equipment shall be able to determine the location of the entire train 7 The indication provided shall enable the driver to drive at the permitted speed without receiving a warning and without intervention of ETCS |
|--|

Gambar 2 Contoh kebutuhan perangkat lunak berbahasa alami [17].

Proses *data cleaning* bertujuan untuk mengidentifikasi dan menghapus data yang salah, rusak, atau tidak terformat dalam kebutuhan perangkat lunak. Langkah ini diperlukan karena dataset yang digunakan dalam penelitian ini berasal dari sumber yang berbeda dengan format pengkodean yang berbeda-beda, sehingga menghasilkan data yang tidak seragam yang dapat mengganggu proses *dataset preprocessing* dan performa model. Gambar 4 adalah salah-satu contoh temuan pada dataset yang digunakan.