

# Implementasi Pemrosesan Bahasa Alami untuk Deteksi Pola Ambiguitas Struktural pada Spesifikasi Kebutuhan Perangkat Lunak ber-Bahasa Indonesia

Chlaudiah Julinar Soplero Lelywiary<sup>1</sup>, Sri Widowati<sup>2</sup>, Kemas Muslim L.<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>chlaudiahjulinar@student.telkomuniversity.ac.id, <sup>2</sup>sriwidowati@telkomuniversity.ac.id,

<sup>3</sup>kemasmuslim@telkomuniversity.ac.id

---

## Abstrak

Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan dokumen yang dihasilkan dari proses rekayasa kebutuhan dan memegang peranan penting dalam pengembangan perangkat lunak. Sekitar 87.7% dokumen SKPL ditulis menggunakan bahasa alami. Masalah terbesar dalam penulisan bahasa alami adalah kesalahan interpretasi yang disebabkan karena terdapat kata-kata yang ambigu. Jika terjadi ambigu dan tidak dideteksi secepat mungkin, maka kesalahan interpretasi dapat mengarah pada hasil perangkat lunak tidak sesuai dengan kebutuhan pengguna. Hal ini membuat masalah ambigu dalam SKPL sangat penting untuk ditangani. Sudah terdapat berbagai penelitian mengenai solusi penanganan ambigu dalam SKPL, dan hampir sebagian besar menggunakan SKPL dalam Bahasa Inggris. Penelitian ini bertujuan untuk mendeteksi ambigu yang terjadi akibat struktur pernyataan kebutuhan perangkat lunak yang salah pada SKPL dalam Bahasa Indonesia. Adapun metode yang diusulkan adalah pola bahasa alami berdasarkan *Part-Of-Speech Tag* Hidden Markov Model-Viterbi, dan pola tersebut dideteksi dengan *Regular Expression Parsing*. Pola bahasa alami yang diusulkan dievaluasi dengan nilai indeks Kappa. Hasil dari analisis pola bahasa alami memiliki nilai indeks Kappa yang tinggi sebesar 0.8481, 0.9139, dan 0.8098, yang berarti ahli sangat sepakat terhadap hasil deteksi ambigu struktural dengan pola bahasa alami *vague words*, ambiguitas analitis, dan ambiguitas koordinasi. Untuk nilai indeks Kappa terendah sebesar -0.0143 yang berarti ahli tidak sepakat terhadap hasil deteksi ambigu struktural dengan pola bahasa alami lampiran preposisi.

**Kata kunci:** spesifikasi kebutuhan perangkat lunak, ambigu, ambigu struktural, pola bahasa alami, *part-of-speech tag*, hidden markov model viterbi, *regular expression parsing*, nilai indeks kappa

---

## Abstract

Software Requirement Specifications (SRS) is a document that is produced from Requirement Engineering (RE) process and plays an important role in software development. About 87.7% of SRS documents are written using natural language. The biggest problem in writing natural language is misinterpretation caused by ambiguous words. If ambiguous happen and not detected as quickly as possible, then misinterpretations can lead to software results that are not in accordance with user needs. This makes the ambiguous problem in SRS very important to deal with. There have been various study on solutions to ambiguous handling in SRS, and most of them use SRS in English. This study aims to detect ambiguity that occurs due to the the structure of incorrect software requirement statements in Indonesian SRS. The proposed method is a natural language pattern based on Part-Of-Speech Tag Hidden Markov Model-Viterbi and the pattern is detected by Regular Expression Parsing. The proposed natural language pattern is evaluated by Kappa index Value. The results of natural language pattern analysis have the high Kappa index value are 0.8481, 0.9139, and 0.8098, which means that experts strongly agree on the results of structural ambiguous detection with natural language textit vague words, analytical ambiguity, and coordination ambiguity. For the lowest Kappa index value of -0.0143 which means the expert disagrees with the results of structural ambiguous detection with the natural language pattern of the preposition attachment.

**Keywords:** software requirement specifications, ambiguous, structural ambiguous, part-of-speech tag, hidden markov model viterbi, regular expression parsing, kappa index value

---

## 1. Pendahuluan

Dalam siklus pengembangan perangkat lunak, memahami, mengumpulkan dan menentukan kebutuhan pengguna terhadap perangkat lunak merupakan tahap awal yang harus dilakukan dan dikenal sebagai proses rekayasa kebutuhan. Hasil dari tahap tersebut berupa dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) yang berisi