

# 1. Pendahuluan

## Latar Belakang

Dalam beberapa periode terakhir, *parsing* telah berkembang pesat meliputi cakupan yang luas, akurasi yang tinggi, dan efisiensi dalam melakukan *parsing* [1]. *Parsing* memiliki beberapa aplikasi, diantaranya adalah *question answering*, *information extraction*, *semantic disambiguation*, *machine translation*, dan *speech synthesis* [2]. Terdapat 2 (dua) teknik utama dalam *parsing*, yaitu *Rule-based* dan *Statistical Parsing* [3] [4]. *Natural Language Processing* (NLP) adalah area dari penelitian interdisipliner dengan tujuan untuk mengembangkan program komputer yang dapat menghasilkan teks dalam bahasa alami [5], yang memungkinkan komputer untuk berkomunikasi dengan manusia dengan cara yang sama seperti bagaimana manusia berkomunikasi dengan manusia lain [6]. *Natural Language Processing* (NLP) melakukan prosesnya dengan beberapa tahapan dan diimplementasikan kedalam suatu aplikasi. Beberapa tahapan tersebut antara lain, yaitu *parsing* yang memeriksa kebenaran suatu struktur kalimat berdasarkan *grammars*, *part of speech tagger* yang memberi penanda POS untuk tiap kata di dalam korpus, *named entity recognizer* yang mendeteksi dan mengklasifikasikan *named-entity* pada suatu teks, *context-independent* yang merepresentasikan arti dari kalimat secara *context-independent*, dan *contextual interpretation* yang untuk merepresentasikan arti secara *context-dependent* [7].

*Parse tree* adalah pohon yang mewakili sintaksis struktur kalimat sehubungan dengan tata bahasa formal, yang dapat dihasilkan secara otomatis oleh *parser* [6]. Diberikan kalimat gramatikal yang benar, *parser* menghasilkan *parse tree* yang sesuai dengan aturan produksi yang ada pada *grammars*. *Parser* begitu penting dalam *Natural Language Processing* (NLP) yang berguna untuk memahami arti dari bahasa manusia dengan menentukan *grammar* dari bahasa tersebut. *Parser* membentuk *parse tree* dari sebuah kalimat, dan bisa diketahui ide utama dari kalimat tersebut [8]. Berbagai aplikasi menggunakan *parser* sebagai sarana pendukung untuk menemukan cara kalimat itu dibentuk dengan pendekatan tata bahasa yang benar dan melibatkan pengetahuan linguistik [9].

Banyak sekali penelitian sebelumnya yang telah berhasil mengubah sebuah kalimat dari berbagai bahasa ke dalam bentuk *parse tree*, sebagai contohnya adalah Maad Shatnawi dan Boumediene Belkhouche yang berhasil mengubah kalimat bahasa Arab ke dalam bentuk *parse tree* dengan menggunakan *Top-Down (Recursive Descent) Parsing* [2] dan Weiwei Sun, Yufei Chen, Xiaojun Wan, dan Meichun Liu yang berhasil mengubah kalimat bahasa Chinese ke dalam bentuk *parse tree* dengan *Grammatical Relations* [10]. Untuk Bahasa Indonesia sendiri, hanya sedikit yang melakukan penelitian terhadap topik ini. Sebagai contoh, terdapat 2 (dua) penelitian yang berhasil mengubah kalimat Bahasa Indonesia ke dalam bentuk *parse tree* dengan menggunakan *Collins's Parser* [6] [11]. Namun 2 (dua) penelitian tersebut tidak dilakukan pengujian performansi, yang berfungsi untuk mengetahui performansi dari pembentukan *parse tree* yang berhasil dibuat oleh sistem tersebut.

Pada penelitian tugas akhir ini, dibangun sebuah sistem untuk mengubah kalimat Bahasa Indonesia ke dalam bentuk *parse tree* dengan menggunakan metode *Bottom-Up Parsing*. *Bottom-Up Parsing* adalah sebuah metode *parsing* berbasis *rule* [12] dengan membentuk *parse tree* dari *leaf (bottom)* lalu ke *root* (akar) [13]. Cara kerja metode ini begitu sederhana, namun performa yang dihasilkan begitu baik dan efisien. Sistem ini melakukan proses *parsing* dengan berdasarkan pada *grammars* yang didapatkan dari proses ekstraksi *treebank*, lalu menghasilkan output berupa *parse tree* dengan bentuk *constituency tree*. Bentuk *parse tree* yang dihasilkan sistem ini dievaluasi pengujian performansi dengan menggunakan *PARSEVAL metrics*.

## Topik dan Batasannya

*Parsing* merupakan langkah penting dalam beberapa aplikasi yang melibatkan analisis dokumen, diantaranya adalah *question answering*, *information extraction*, *semantic disambiguation*, *summarization*, dan *filtering* [2]. *Parser* membentuk *parse tree* dari sebuah kalimat, dan bisa diketahui ide utama dari kalimat tersebut [8]. Berbagai aplikasi menggunakan *parser* sebagai sarana pendukung untuk menemukan cara kalimat itu dibentuk dengan pendekatan tata bahasa yang benar dan melibatkan pengetahuan linguistik [9].

Dalam rangka sarana pendukung berbagai aplikasi pengolah bahasa guna meningkatkan penelitian di bidang *Natural Language Processing* (NLP), maka *parser* bahasa Indonesia diperlukan dan perlu dikembangkan keberadaannya. Untuk memperkecil cakupan dari tugas akhir ini diperlukan batasan-batasan. Adapun batasan masalah yang diambil untuk tugas akhir ini adalah sebagai berikut:

- *Output* sistem hanya menampilkan 1 (satu) bentuk *parse tree* paling atas (dengan kemungkinan paling benar), jika suatu kalimat menghasilkan lebih dari 1 (satu) *parse tree*.
- Hanya bisa melakukan *parsing* jika kata (lexicon) dalam kalimat terdapat pada *grammars*. Jika kata (lexicon) dalam kalimat tidak terdapat pada *grammars*, sistem menghasilkan output berupa *message* tidak dapat melakukan *parsing* karena kata pada kalimat tidak ada pada *grammars*.

**Tujuan**

Tujuan dibuatnya sistem ini adalah untuk menghasilkan bentuk *parse tree* dari sebuah kalimat Bahasa Indonesia menggunakan metode *Bottom-Up Parsing*, kemudian mengevaluasi bentuk *parse tree* yang telah dibuat oleh sistem dengan menghitung nilai *precision*, *recall*, dan *f1 score* menggunakan pengukuran *PARSEVAL metrics*.

**Organisasi Tulisan**

Penulisan tugas akhir ini dibagi menjadi 5 (lima) bagian, yaitu:

1. Bagian 1 berisi latar belakang, topik dan batasannya, tujuan, dan organisasi tulisan.
2. Bagian 2 berisi topik-topik terkait dengan tugas akhir ini.
3. Bagian 3 berisi alur sistem yang dibangun pada tugas akhir ini.
4. Bagian 4 berisi hasil pengujian sistem yang sudah dibangun beserta analisis terkait hasil yang sudah ada.
5. Bagian 5 berisi kesimpulan dari seluruh bagian penelitian tugas akhir yang sudah dilakukan. Terdapat saran untuk mengembangkan sistem ini di kemudian hari.