**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : SA`AD AHYAT HASAN**

**NRP : 5108100146**

**DOSEN WALI : TOHARI AHMAD, S.Kom., MIT., Ph.D.**

# JUDUL TUGAS AKHIR

Rancang Bangun API *Part of Speech Tagger* untuk Bahasa Indonesia Menggunakan Hidden Markov Model.

*Software Design of Part of Speech Tagger for Bahasa Indonesia based on Hidden Markov Model.*

# ABSTRAKSI

Pada tugas akhir ini akan dibuat API penanda kelas kata (*part-of-speech tagger*) bahasa Indonesia dengan menggunakan metode *Hidden Markov Model*.  
*Hidden Markov Model* (HMM) merupakan model statistik dimana sistem yang dimodelkan diasumsikan sebagai proses Markov dengan parameter yang tidak diketahui dan kemudian menentukan parameter yang tidak diketahui tersebut dengan parameter terobservasi sesuai dengan asumsi yang ada.

Selain *Hidden Markov Model*, beberapa metode lain yang seringkali digunakan untuk membuat *part-of-speech tagger* adalah: *Maximum Entropy*, *Conditional Random Field*, dan juga pendekatan berbasis transformasi. Dari sekian banyak metode, *Hidden Markov Model* terbukti mempunyai *running time* yang paling baik.

Meskipun sudah ada beberapa penelitian tentang *part-of-speech tagger* untuk bahasa Indonesia, masih belum ditemukan API *part-of-speech tagger* untuk bahasa Indonesia yang bisa digunakan secara luas. Hal itu menyebabkan setiap peneliti di bidang pemrosesan bahasa alami untuk bahasa Indonesia harus membuat *part-of-speech tagger* sendiri. Dan hal itu sangat tidak efektif dan efisien.

Tujuan dari tugas akhir ini yaitu untuk memudahkan pemrosesan kalimat dalam bahasa Indonesia. Sehingga pihak-pihak yang ingin melakukan penelitian di bidang pemrosesan bahasa alami untuk bahasa Indonesia tidak harus membuat *part-of-speech tagger* sendiri dari awal. Mengingat *part-of-speech tagging* merupakan salah satu proses awal dari serangkaian proses yang ada dalam pemrosesan bahasa alami

# PENDAHULUAN

# LATAR BELAKANG

Bahasa merupakan sebuah instrumen yang paling penting dalam berkomunikasi. Tak terkecuali bagi komputer. Seperti yang kita tahu, komputer hanya mampu mengenali bahasa mesin. Di lain sisi, tidak banyak manusia yang mampu memahami bahasa mesin. Dari sana kemudian muncul ide tentang pemrosesan bahasa alami atau yang lebih umum dikenal sebagai NLP (*Natural Language Processing*) yang memungkinkan komputer untuk dapat memahami bahasa manusia. Sehingga memungkinkan munculnya teknologi-teknologi baru seperti pengenalan suara, penerjemah bahasa, dan lain sebagainya.

Pemrosesan bahasa alami merupakan sebuah cabang ilmu komputer dan linguistik yang mengkaji interaksi antara komputer dengan bahasa (alami) manusia. Pemrosesan bahasa alami sering dianggap sebagai cabang dari kecerdasan buatan dan bidang kajiannya bersinggungan dengan linguistik komputasional. Kajian pemrosesan bahasa alami antara lain mencakup segmentasi tuturan (*speech segmentation*), segmentasi teks (*text segmentation*), pengawataksaan makna (*word sense disambiguation*), serta penandaan kelas kata (*part-of-speech tagging*).

*Part-of-speech tagging* adalah proses penandaan kata pada suatu teks (korpus) dalam kaitannya dengan suatu kelas kata tertentu berdasarkan definisi dan maknanya serta hubungannya dengan kata yang mendampingi atau yang terkait dengannya pada suatu frasa, kalimat, atau paragraf. Bentuk sederhana proses ini adalah identifikasi kata sebagai nomina, verba, adjektiva, adverbia, dll.

Khusus untuk *part-of-speech tagging* dalam bahasa Indonesia, sudah ada beberapa penelitian yang dilakukan dengan menggunakan berbagai macam metode dan pendekatan. Pisceldo dkk. [[1](#Fem09)] mengembangkan *part-of speech tagger* untuk bahasa Indonesia menggunakan Maximum Entrophy model dan juga *Conditional Random Field* (CRF). Triastuti [[2](#Ind)] mengembangkan *part-of speech tagger* untuk bahasa Indonesia menggunakan CRF, pendekatan berbasis tranformasi, dan juga gabungan antara CRF dan pendekatan berbasis transformasi. Sari dkk. [[3](#Sar08)] juga menggunakan pendekatan transformasi Brill berbasis aturan. Sedangkan Alfan dkk. [[4](#Alf10)] mengembangkan *part-of speech tagger* untuk bahasa Indonesia menggunakan metode *Hidden Markov Model* (HMM). Dari semua metode yang telah disebutkan tersebut, metode HMM terbukti mempunyai *running time* yang paling baik [[5](#Nam07)].

Meskipun sudah ada beberapa penelitian mengenai *part-of speech tagger* untuk bahasa Indonesia, sampai saat ini belum ada API (*Application Programming Interface*) *part-of speech tagger* bahasa Indonesia yang bisa digunakan secara luas. Padahal keberadaan API *part-of speech tagger* bahasa Indonesia dapat membantu perkembangan penelitian di bidang pemrosesan bahasa alami dalam bahasa Indonesia.

Oleh karena itu, dalam TA ini akan dikembangan sebuah API *part-of speech tagger* bahasa Indonesia dengan menggunakan metode *Hidden Markov Model* (HMM). Dengan harapan agar nantinya dapat membantu perkembangan penelitian di bidang pemrosesan bahasa alami untuk bahasa Indonesia. Sehingga ke depan, para peneliti di bidang pemrosesan bahasa alami untuk bahasa Indonesia tidak lagi harus membuat *part-of speech tagger* secara manual dari awal.

# RUMUSAN MASALAH

Permasalahan-permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut:

* 1. Bagaimana caranya membuat korpus bahasa Indonesia?
  2. Bagaimana menggunakan metode Hidden Markov Model dalam pembuatan *part-of speech tagger* untuk bahasa Indonesia?
  3. Bagaimana membuat API (*Application Programming Interface*)?

# BATASAN MASALAH

Batasan masalah dari tugas akhir ini adalah:

* + 1. Tagset bahasa Indonesia yang digunakan adalah tagset yang mempunyai kesamaan dengan tagset bahasa Inggris yang ada dalam *The Penn Treebank Tagset*.
    2. Menggunakan bahasa pemrograman Java.
    3. API akan dibuat dalam bentuk .jar.

# TUJUAN DAN MANFAAT PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan tugas akhir ini adalah:

1. Mengembangkan aplikasi pustaka penanda kelas kata untuk bahasa Indonesia yang bisa digunakan secara luas.
2. Meningkatkan efektivitas waktu dalam penelitian di bidang pemrosesan bahasa alami untuk bahasa Indonesia.

# TINJAUAN PUSTAKA

## Rancang Bangun Perangkat Lunak

Rancang bangun perangkat lunak merupakan tahap-tahap teknis untuk membangun perangkat lunak yang melingkupi analisis permasalahan dan kebutuhan, perencanaan, analisis sistem, implementasi, serta aktivitas pengujian dan pemeliharaan perangkat lunak.

Rancang bangun perangkat lunak diperlukan untuk menentukan konsep, strategi, dan praktik yang baik diterapkan untuk menciptakan perangkat lunak yang berkualitas tinggi, sesuai anggaran biaya, mudah dalam pemeliharaannya, serta tidak membutuhkan waktu yang lama dalam pembangunannya. Beberapa model rancang bangun perangkat lunak yang terkenal dan banyak dipakai antara lain: model air terjun dan model iterasi .

## Korpus

Korpus atau korpora adalah sejenis bank bahasa yang berasal dari penggunaan bahasa dalam berbagai genre dan ragam, baik lisan maupun tertulis yang menjamin keragaman yang seluas-luasnya dan menghindari penggunaan bahasa yang sangat sempit seperti idiolek. Data tersebut disusun secara sistematis dan biasanya mudah diakses secara elektronis dengan komputer [[7](#Kri08)].

* 1. **API**

API merupakan sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi [[8](#Wik13)].

* 1. **Hidden Markov Model**

Hidden Markov Model merupakan model statistic dimana suatu sistem yang dimodelkan diasumsikan sebagai kondisi yang tidak terobservasi. Suatu HMM dapat dianggap sebagai jaringan Bayesian dinamis yang sederhana. (*simplest dynamic Bayesian network*) [[9](#Pra10)].

# METODOLOGI

Berikut langkah-langkah pengerjaan yang digunakan dalam pembuatan tugas akhir ini:

* + - 1. **Mencari kebutuhan fungsional dan non-fungsional dari tugas akhir.**

Kebutuhan fungsional dan non fungsional didasarkan pada hasil pengamatan dari aplikasi-aplikasi *part-of-speech* *tagger* untuk bahasa inggris seperti yang ada pada stanfordNLP dan juga openNLP. Dengan begitu, diharapkan *part-of-speech tagger* yang dibuat nantinya akan memenuhi kebutuhan pengguna.

* + - 1. **Studi literatur yang dibutuhkan untuk kebutuhan yang telah ditemukan.**

Setelah kebutuhan fungsional dan non-fungsional diketahui, langkah selanjutnya adalah mencari solusi atas kebutuhan-kebutuhan yang ada. Beberapa literatur yang akan digunakan dalam proses studi literatur ini diantaranya adalah: dokumentasi program-program *part-of-speech tagger* yang sudah dikembangkan sebelumnya, buku, jurnal, paper, dan juga artikel-artikel ilmiah yang ada di internet yang berkaitan dengan *part-of-speech tagging* dan API.

* + - 1. **Merancang desain, alur proses, dan algoritma yang digunakan dalam program.**

Proses desain baru akan dilakukan berdasarkan hasil studi literatur. Rancangan desain yang akan dibuat dalam tugas akhir ini nantinya meliputi: *architecture system diagram*, *use case diagram*, dan juga *class diagram*.

* + - 1. **Melakukan implementasi berdasarkan desain yang sudah dibuat.**

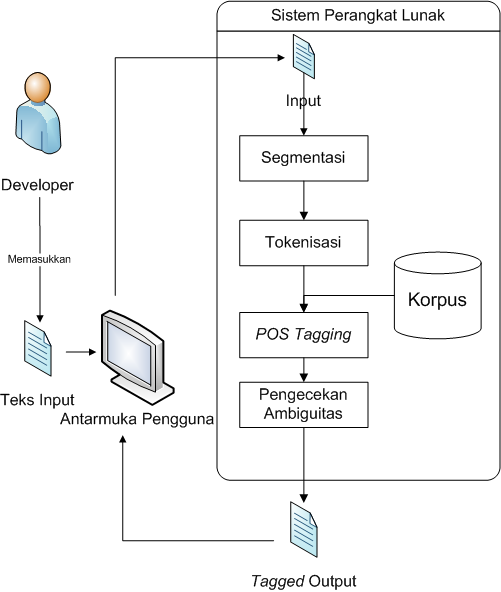
Secara umum, proses implementasi ini akan terdiri dari 2 macam proses utama. Yang pertama adalah proses pembuatan korpus. Dan yang kedua adalah proses pembuatan *part-of-speech tagger*. Untuk *part-of-speech tagger* akan diimplementasikan dengan menggunakan bahasa pemrograman java dengan mengacu pada desain sistem yang sudah dibuat pada proses sebelumnya. Untuk desain awal program akan dijelaskan kemudian.

* + - 1. **Uji coba hasil program yang telah dibuat.**

Sama halnya dengan proses implementasi, proses uji coba juga terdiri dari 2 sub proses. Yang pertama adalah uji coba kualitas dan tingkat akurasi hasil *part-of-speech tagger*. Dan yang kedua adalah uji coba kinerja API *part-of-speech tagger*.

* + - 1. **Revisi kesalahan dan *bug* yang muncul.**
      2. **Demo akhir.**

Berikut ini desain awal dari aplikasi *part-of-speech tagger* yang akan dibuat:



Gambar 1. Arsitektur Sistem

Dari gambar 1 dapat diketahui bahwa sistem terdiri dari 4 proses utama yakni: segmentasi, tokenisasi, *pos tagging*, dan pengecekan ambiguitas.

**Segmentasi**

Segmentasi teks adalah proses pemisahan teks tertulis menjadi unit makna seperti kata, kalimat, atau topik. Istilah ini dapat diterapkan baik untuk proses mental yang dilakukan oleh manusia sewaktu membaca teks, maupun proses buatan yang dilakukan oleh komputer dan menjadi bahan kajian pemrosesan bahasa alami. Meskipun beberapa aksara memiliki batas kata eksplisit (seperti spasi) atau pembedaan bentuk huruf awal, tengah, dan akhir (seperti pada aksara Arab), penanda tersebut kadang taksa dan tidak semua bahasa tulisan memilikinya.

**Tokenisasi**

Tokenisasi adalah proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token. Sebagai contoh, tokenisasi dari kalimat "Aku baru saja makan bakso pedas." menghasilkan enam token, yakni: "Aku", "baru", "saja", "makan", "bakso", "pedas". Biasanya, yang menjadi acuan pemisah antar token adalah spasi dan tanda baca. Tokenisasi berguna untuk analisis teks lebih lanjut dan dipakai dalam ilmu linguistik.

**POS Tagging**

*POS tagging* merupakan proses penandaan kelas kata dari kalimat-kalimat yang sudah dipisahkan menjadi token-token.

**Pengecekan Ambiguitas**

Pengecekan ini untuk memastikan kelas kata dari kata-kata yang homograf. Untuk membantu proses ini, akan digunakan analisis leksikal berdasarkan pada KBBI dan kateglo (kamus, tesaurus, dan glosarium).



Gambar 2. Use Case Diagram

Dari gambar 2 dapat diketahui bahwa setidaknya ada 5 fitur yang bisa digunakan oleh user dari sistem ini. 5 fitur tersebut merupakan jumah fitur minimal dan setiap fitur diwakili oleh sebuah kelas dalam API. Dan jika mengacu pada *part-of-speech* yang ada di [[10](#Sta)], setidaknya ada lebih dari 30 kelas yang bisa digunakan oleh user. Namun, mengingat ada perbedaan antara kaidah dan tata bahasa antara bahasa Inggris dengan bahasa Indonesia, sangat dimungkinkan jumlah kelas yang ada dalam *part-of-speech tagger* untuk bahasa Indonesia ini nanti tidak akan sama persis dengan yang ada pada aplikasi *part-of-speech tagger* untuk bahasa Inggris.

# JADWAL PEMBUATAN TUGAS AKHIR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | 2012 | | | | | | | | | | | | | | | | |
| Maret | | | | April | | | | Mei | | | | Juni | | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

x

|  |  |
| --- | --- |
| [1] | Femphy Pisceldo, Manurung, R.Adriani, Mirna, "Problabilistic Part-of-Speech Tagging for bahasa Indonesia," in *Third International MALINDO Workshop, colocated event ACL-IJCNLP 2009*, Singapore, 2009. |
| [2] | Triastuti Chandrawati, *Indonesian Part-of-Speech Tagger based on Conditional Random Field and Transformation based learning methods*, Undergraduated thesis., Ed. Depok: Fasilkom University of Indonesia., 2008. |
| [3] | Sari, Syandra, Herika Haruyani, Mirna Adriani, and Stephen Bressan., "Developing Part-of-Speech Tagger for Bahasa Indonesia Using Brill Tagger," in *The International Second MALINDO Workshop*, 2008. |
| [4] | Alfan Farizki Wicaksono, Ayu Purwarianti, "HMM Based POS Tagger for Bahasa Indonesia," in *On Proceedings of 4th International MALINDO (Malay - Indonesian Language) Workshop*, 2010. |
| [5] | Nam Nguyen. Yunsong Guo, "Comparisons of Sequence Labeling Algorithm and Extensions," in *International Conference on Machine Learning*, 2007. |
| [6] | Ian Sommerville, *Software Engineering*. Boston: Addison-Wesley, 2007. |
| [7] | Harimurti Kridalaksana, *Kamus Linguistik*, 4th ed. Jakarta: Gramedia Pustaka Utama, 2008. |
| [8] | (2013, March) Wikipedia. [Online]. <http://en.wikipedia.org/wiki/Application_programming_interface> |
| [9] | Muhammad Eko Budi Prasetyo, Teori Dasar Hidden Markov Model, April 2010. |
| [10] | Standford NLP. [Online]. <http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/package-summary.html> |

x