

Implementasi Algoritma Knuth Morris Pratt pada Alat Penerjemah Suara

Bima Laksmna Pramudita (13511042)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13511042@std.stei.itb.ac.id

Abstrak—Alat penerjemah suara, merupakan alat untuk mengenali suara percakapan dalam suatu bahasa yang diberikan dan dapat menerjemahkannya menjadi percakapan dalam suatu bahasa yang berbeda. Alat ini sudah bisa diintegrasikan ke dalam mobile phone sehingga mudah untuk digunakan. Begitu banyak metode yang digunakan untuk pembuatan alat penerjemah tersebut, salah satunya dengan menggunakan algoritma Knuth Morris Pratt (KMP), dan tentunya dengan piranti lainnya yang dapat mendukung seperti *Speech Recognition* yang akurat sehingga dapat memaksimalkan hasil yang diterima. Pada *Translation Software* atau *Machine Translation*, algoritma KMP ini akan meninjau kata-kata dari setiap teks yang akan diterjemahkan kemudian dicari kecocokannya dengan kata yang sebanding dalam bahasa lain yang diinginkan dengan menggunakan metode yang kompleks sehingga menghasilkan . Dalam makalah ini penulis akan melakukan analisa terhadap implementasi algoritma KMP dalam alat penerjemah suara (*Voice Translation Device*) dan juga akan sedikit menyentuh kinerja *Speech Recognition* yang digunakan dalam alat tersebut.

Kata Kunci—penerjemah, KMP, suara, teks.

I. PENDAHULUAN

Alat penerjemah suara telah dikembangkan sekitar pada 15 tahun yang lalu. Sudah banyak prototype dan hasil yang menjanjikan pada alat tersebut. Membayangkan bahwa suatu hari semua orang bisa bepergian ke semua negara di dunia, dan bisa berbincang percakapan dengan orang yang berbicara menggunakan bahasa yang berbeda, hanya dengan menggunakan sebuah alat berukuran kecil dan bisa dibawa ke semua tempat.

Pada saat ini, alat penerjemah suara telah digunakan di seluruh dunia. Fasilitas medis, sekolah, hotel, toko, dan juga pabrik telah menggunakan sistem ini.

Dalam alat penerjemah suara ini, akurasi dan

kecepatan dalam memproses data *string* merupakan hal yang krusial dalam mengembangkan alat semacam ini. Seorang user dapat menginput data berupa suara dari sebuah kalimat yang akan diterjemahkan ke dalam bahasa yang diinginkan, kemudian secara otomatis alat penerjemah suara akan mengeluarkan output sebuah terjemahan kalimat dalam bentuk suara dengan bahasa yang diinginkan sebelumnya

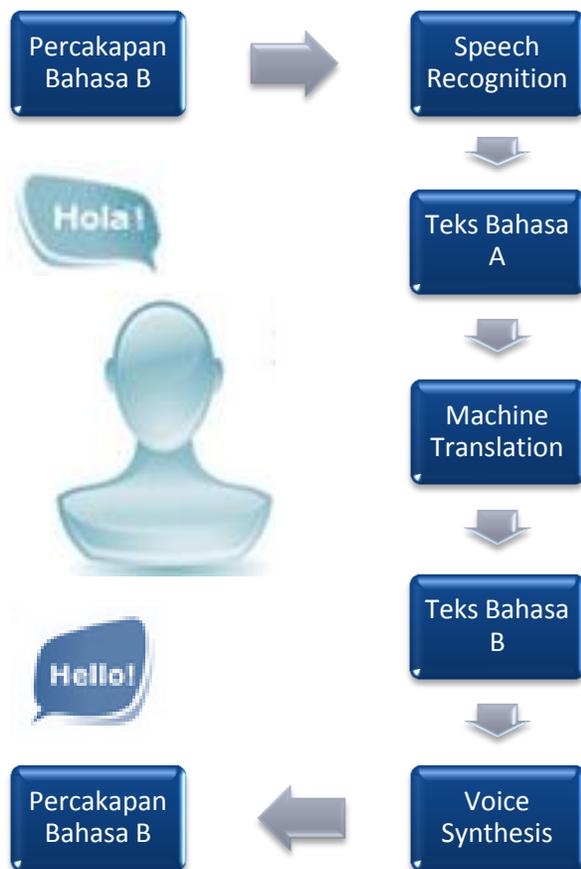
Pada umumnya, alat penerjemah suara bekerja dengan menganalisa input suara yang diberikan, kemudian mengubahnya menjadi sebuah kata *string*. Dengan melakukan *string matching*, input yang telah diubah menjadi kata *string* akan dicocokkan berdasarkan kata-kata yang berada di dalam database yang merupakan kamus alat tersebut. Kemudian mengembalikan output suara dari hasil terjemahan kata-kata tersebut.

Pencocokan *string* pada saat menerjemahkan kata dapat dilakukan dengan berbagai algoritma, seperti Brute force, Knuth Morris Pratt (KMP), Boyer Moore, dan lainnya. Namun algoritma yang bisa dibidang cukup mangkus yakni algoritma Knuth Morris Pratt.

Algoritma Knuth Morris Pratt atau disingkat KMP adalah algoritma pencarian atau pencocokan *string* dengan mengingat beberapa perbandingan yang dilakukan sebelumnya, sehingga dapat meningkatkan besar pergeseran yang dilakukan. Hal tersebut bisa menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian.

II. ALAT PENERJEMAH SUARA

Alat penerjemah suara merupakan suatu sistem yang terintegrasi dengan beberapa teknologi: *Automatic Speech Recognition* (ASR), *Machine Translation* (MT), dan *Voice Synthesis* (TTS).



Gambar 1. Cara Kerja Alat Penerjemah Suara

Pembicara dari bahasa A berbicara ke sebuah microphone dan modul speech recognition akan mengenali ucapan. Kemudian membandingkan input tersebut dengan model fonologi, yang terdiri dari sebuah kumpulan tulisan dari data percakapan dari beberapa pembicara. Input tersebut kemudian diubah menjadi kumpulan kata-kata *string*, dengan menggunakan kamus dan tatabahasa dari bahasa A yang didasari dari kumpulan tulisan dari bahasa A.

Kata *string* yang telah terkumpul kemudian diterjemahkan dengan menggunakan modul *Machine Translation*. Pada awalnya, sistem mengganti semua kata dengan kata yang sesuai dengan pada bahasa B, dimana bahasa B adalah bahasa yang diinginkan untuk hasil terjemahan. Sistem pada saat ini tidak menggunakan translasi kata demi kata, melainkan memperhitungkan seluruh konteks input sehingga sistem tersebut dapat menghasilkan terjemahan yang sesuai.

Terjemahan ucapan yang telah dihasilkan dikirim ke modul *Speech Synthesis* yang memperkirakan

pengucapan dan intonasi yang sesuai dengan serangkaian kata-kata dari sebuah data kumpulan tulisan dalam bahasa B yang kemudian akan dikeluarkan oleh alat penerjemah suara dalam bentuk suara.



Gambar 2. Contoh Alat Penerjemah Suara

III. SPEECH RECOGNITION

Speech Recognition merupakan translasi dari sebuah kata yang diucapkan menjadi teks, yang biasa dikenal dengan "*Automatic Speech Recognition*", "*ASR*", "*Computer Speech Recognition*", "*Speech To Text*", atau "*STT*".

Ada dua pemodelan dasar untuk speech recognition, yakni *Hidden Markov model (HMM)-Based Speech Recognition* dan *Dynamic Time Warping (DTW)-Based Speech Recognition*.

Modern general-purpose speech recognition system umumnya menggunakan model *Hidden Markov*. Satu alasan untuk menggunakan *Hidden Markov* karena sebuah sinyal dari sebuah ucapan bisa dilihat seperti *piecewise stationary signal* atau *short-time stationary signal* dan alasan lainnya mengapa metode ini digunakan yaitu sederhana dan secara komputasional bisa digunakan.

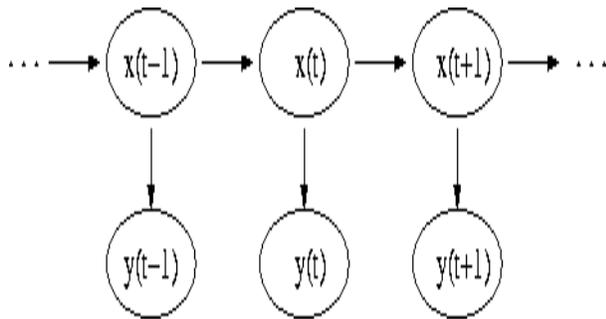
Secara garis besar *Hidden Markov Model* merupakan model yang statistik dari sebuah sistem yang diasumsikan sebuah proses Markov dengan parameter yang tidak diketahui, dan tantangannya adalah menentukan parameter-parameter tersembunyi (state) dari parameter-parameter yang dapat diamati (observer).

Dalam penggunaan *Hidden Markov Model*, ada 2 permasalahan yang dapat diselesaikan oleh HMM :

Pertama, diberikan parameter dari model, perhitungan probabilitas output berupa suatu barisan

tertentu, perhitungan tersebut dapat diselesaikan oleh *forward algorithm*.

Kedua, pencarian barisan state tersembunyi yang paling mungkin menghasilkan output barisan tertentu saat diberikan parameter dari model, pencarian tersebut dapat diselesaikan dengan algoritma Viterbi. Ketiga, diberikan sebuah barisan output atau himpunan barisan, maka pencarian himpunan transisi state yang paling mungkin beserta probabilitas outputnya., pencarian tersebut dapat diselesaikan dengan algoritma Baum-Welch.



Gambar 3. Diagram Arsitektur HMM

Dynamic Time Warping adalah algoritma untuk menghitung *optimal warping path* antara dua waktu. Algoritma ini menghitung baik antara nilai *warping path* dari dua waktu dan jaraknya.

Algoritma ini digunakan untuk speech recognition sebelum diganti dengan *Hidden Markov Model* (HMM).

Dynamic Time Warping sangat berguna untuk mengisolasi pengenalan kata yang diucapkan dalam kamus yang bisa dikatakan terbatas, sehingga kompleksitasnya tidak memuaskan untuk kamus yang lebih besar. Algoritma *Dynamic Time Warping* sulit mengevaluasi dua elemen dari dua sekuens yang berbeda, mengambil dari nilai yang ada banyak jalur yang memiliki fitur khusus.

IV. ALGORITMA KNUTH MORRIS PRATT

Algoritma Knuth Morris Pratt merupakan salah satu algoritma pencarian *string*, algoritma ini memelihara informasi yang digunakan untuk melakukan jumlah pergeseran pada setiap kali ditemukan ketidakcocokan pattern dengan teks. Algoritma ini menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh, tidak hanya satu karakter. Dengan menggunakan algoritma Knuth

Morris Pratt, waktu pencarian dapat dikurangi secara signifikan.

Proses dengan menggunakan algoritma KMP akan seperti berikut:

Misalkan pattern = "abcabd" akan dicari pada sebuah teks="abcabcabd", prosesnya adalah sebagai berikut.

a	b	c	a	b	c	a	b	d
a	b	c	a	b	d			

Pencocokan pertama yaitu samakan (align) ujung kiri pattern dengan ujung kiri teks. Karakter-karakter pada posisi 1..5 sama (match), tetapi karakter c dan d pada posisi 6 menghasilkan ketidakcocokan. Untuk menentukan jumlah pergeseran yang akan dilakukan, mula-mula harus dilakukan perhitungan fungsi pinggiran untuk karakter yang sama atau cocok. Fungsi pinggiran adalah sebagai ukuran awal terpanjang dari suatu *string P* yang merupakan akhiran dari $P[1..j]$.

Fungsi pinggiran untuk pattern "abcabd" akan didefinisikan sebagai berikut:

j	1	2	3	4	5	6
$P[j]$	a	b	c	a	b	D
$B(j)$	0	0	0	1	2	0

Jumlah pergeseran selanjutnya ditentukan oleh fungsi pinggiran dari awalan P yang bersesuaian (match). Dalam kasus ini, awalan yang bersesuaian adalah "abcab", yang memiliki panjang 5. Dari fungsi pinggiran diatas dapat dilihat bahwa pinggiran yang terpanjang untuk *string P*[1..5] adalah "ab" yang panjangnya bernilai 2 karakter. Maka berdasarkan fungsi pinggiran diatas, jarak pergeseran adalah $5 - 2 = 3$. Jadi pattern P digeser sejauh 3 karakter dan perbandingan dilakukan mulai dari posisi $j = 3$ yang dihitung dari awal pattern.

a	b	c	a	b	c	a	b	d
			a	b	c	a	b	d

Setelah dilakukan pergeseran yang berdasarkan fungsi pinggiran, maka pada akhirnya ditemukan kecocokan dalam *string* diatas.

Sebagai contoh lain, terdapat kata “pagi” yang ingin dicari kata yang sebanding dalam bahasa inggris. *Machine Translation* akan melakukan pencocokan kata untuk mencari kata yang sebanding tersebut dalam bahasa asing. *Machine Translation* juga akan memperhitungkan apakah kata tersebut adalah bagian dari sebuah kata majemuk yang lain. Misalkan kata “selamat” yang dimaksud dalam hal ini merupakan bagian dari “selamat pagi” yang dapat dianggap sebagai satu kesatuan kata majemuk.

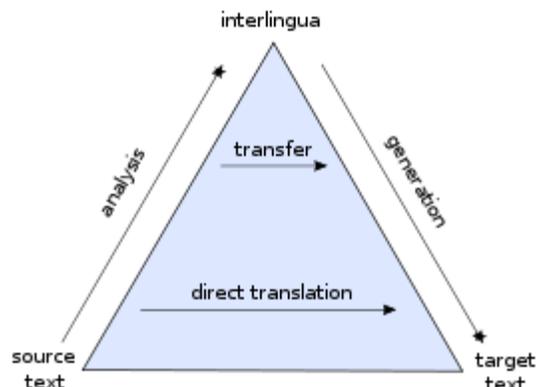
Jika kata “selamat pagi” diterjemahkan kedalam bahasa inggris kata demi kata, maka akan terjemahan dari kata “selamat” adalah “congratulations” dan terjemahan dari kata “pagi” adalah “morning. Jika kedua hasil terjemahan disatukan, maka akan menghasilkan sebuah kata “congratulations morning” dimana kata tersebut tidak tepat dalam arti dalam bahasa inggris.

Terjemahan kata demi kata telah dibuktikan tidak dapat menerjemahkan sebuah kata majemuk atau kalimat secara benar, maka diperlukan proses yang lebih kompleks lainnya seperti peninjauan terhadap idiom ataupun pola kalimat lainnya, walau secara umum *Machine Translation* ini melakukan translasi atau penerjemahan dengan melakukan *string matching*.

Hal yang dilakukan *Machine Translation* dalam proses translasi didasari oleh 2 hal :

1. *Decoding* makna dari sebuah sumber teks.
2. *Re-encoding* makna tersebut ke bahasa target.

Untuk *Decode* seluruh makna dari sebuah sumber teks, alat penerjemah harus menginterpretasikan dan menganalisa semua fitur dari teks tersebut, sebuah proses yang memerlukan pengetahuan yang mendalam tentang tatabahasa, semantic, sintaksis, idiom, dan lainnya dari bahasa sumber teks, dan juga budaya dari pembicaranya. Alat penerjemah juga membutuhkan pengetahuan yang mendalam yang sama untuk *Re-encoding* makna tersebut ke bahasa yang diinginkan.



Gambar 4. Pendekatan *Machine Translation* Dalam Menerjemahkan Sebuah Teks

V. KESIMPULAN

Alat penerjemah suara merupakan alat yang penting di masa depan.. Keakuratan dalam menerjemahkan suatu kalimat merupakan hal yang sangat penting dalam mengembangkan suatu alat penerjemah. Selain keakuratan terjemahan, kemangkusan dalam kinerja pemrosesan pada alat penerjemah suara juga merupakan hal yang harus diperhatikan, sehingga user dapat dengan mudah mengoprasikannya serta mendapat hasil terjemahan yang akurat dan hasil yang cepat.

Keakuratan pada semua alat penerjemah (*Machine Translation*) mungkin belum ada yang dapat dikatakan sangat akurat, mengingat pengetahuan yang mendalam tentang setiap bahasa bisa dikatakan sangat berbeda. Tidak ada cara yang terbaik, tetapi selalu ada cara yang lebih baik. *Machine Translation* mungkin belum dapat mengalahkan terjemahan yang dilakukan oleh manusia, tetapi alat penerjemah suara dapat melakukan kerja dengan baik untuk membantu kinerja manusia.

REFERENSI

- [1] Munir, Rinaldi, Diktat Kuliah IF2211 Strategi Algoritma. Program studi Teknik Informatika Intitut Teknologi Bandung, 2009.
- [2] Hutchins, W John; Somers, Harold L. (1992). An Introduction to *Machine Translation*. London: Academic Press.
- [3] <http://www.statmt.org/>

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2013

A handwritten signature in black ink, appearing to be 'Bima Laksmna Pramudita', written in a cursive style.

Bima Laksmna Pramudita
13511042