

Penerapan Algoritma Knuth Morris Pratt dalam Aplikasi Penerjemah Teks

Okharyadi Saputra (13510072)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
okharyadi@gmail.com

Abstrak—Aplikasi penerjemah teks, seperti layaknya Google Translate ataupun Bing Translator merupakan aplikasi yang memiliki kemampuan untuk mengenali teks yang diberikan dalam suatu bahasa dan menerjemahkannya ke dalam bahasa yang lain. Begitu banyak metode algoritma yang dapat digunakan dalam pembuatan aplikasi penerjemah seperti itu, salah satunya dengan menggunakan algoritma Knuth Morris Prath (KMP). Dengan algoritma KMP ini, setiap teks yang akan diterjemahkan akan ditinjau kata-kata untuk kemudian dicari kecocokannya dengan kata padanan dalam bahasa lain yang diinginkan. Dalam makalah ini penulis akan melakukan analisa sekaligus uji coba terhadap penerapan algoritma KMP ini dalam aplikasi penerjemah teks.

Kata Kunci—penerjemah, KMP, teks.

I. PENDAHULUAN

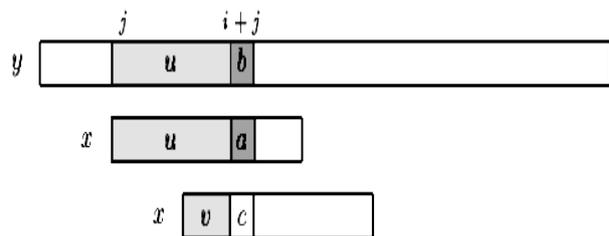
Aplikasi penerjemah teks merupakan aplikasi yang sangat populer saat ini. Didukung dengan kemunculan berbagai aplikasi penerjemah seperti Google Translate, semakin memberikan kemudahan kepada pengguna internet agar dapat menerjemahkan teks yang mereka inginkan ke dalam bahasa apapun. Akurasi sekaligus kecepatan dalam proses penerjemahan merupakan kunci dalam keberhasilan aplikasi semacam ini.

Dalam aplikasi penerjemah teks, seorang *user* dapat menginputkan data berupa teks yang akan diterjemahkan. Selanjutnya, user dapat menentukan teks tersebut akan diterjemahkan ke dalam bahasa apa, dan secara otomatis aplikasi akan menghasilkan output berupa teks dengan hasil terjemahannya.

Pada dasarnya, suatu aplikasi penerjemah teks bekerja dengan melakukan *string matching* atau pencocokan string berdasarkan kata-kata yang terdapat dalam database yang merupakan kamus aplikasi tersebut. Seperti yang kita ketahui, pencocokan string dapat dilakukan dengan berbagai algoritma yang ada, seperti Brute Force dan Knuth Morris Pratt. Sayangnya, algoritma Brute Force tidaklah cukup efisien jika digunakan dalam aplikasi penerjemah teks seperti itu. Terdapat algoritma lain yang lebih mangkus yaitu algoritma Knuth Morris Pratt.

Algoritma Knuth Morris Pratt atau disingkat KMP merupakan algoritma pencocokan atau pencarian string yang merupakan pengembangan dari algoritma Brute

Force. Berbeda dengan algoritma Brute Force yang melakukan pencocokan string dengan pergeseran satu per satu karakter, algoritma KMP mampu melakukan pencocokan dengan pergeseran yang lebih baik.



Gambar 1 Simulasi Kerja Algoritma KMP

II. APLIKASI PENERJEMAH TEKS

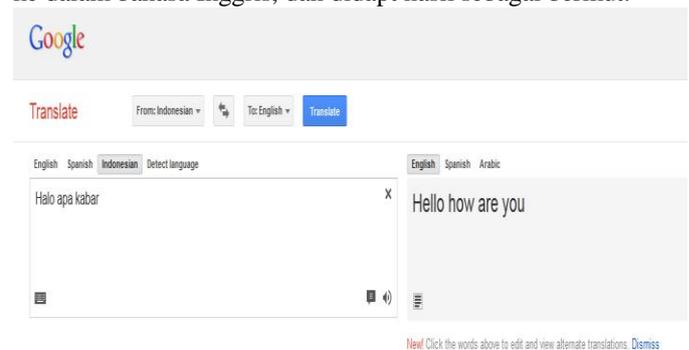
Sebelum penulis melakukan pembahasan lebih lanjut mengenai implementasi algoritma KMP dalam aplikasi penerjemah teks, sebelumnya penulis akan melakukan sedikit peninjauan dan analisa terhadap aplikasi penerjemah yang sudah ada, yaitu Google Translate.

Aplikasi Google Translate merupakan aplikasi penerjemah teks berbasis online yang dapat menerjemahkan teks dalam 65 bahasa yang berbeda, termasuk bahasa Indonesia.

Sebagai contoh, penulis mencoba untuk menginputkan kalimat berbunyi:

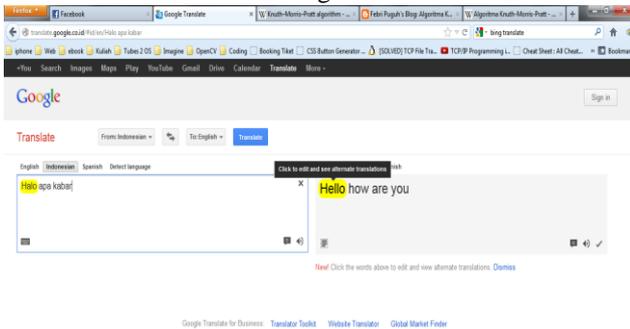
“Halo apa kabar?”

Selanjutnya penulis mencoba untuk menerjemahkannya ke dalam bahasa Inggris, dan didapat hasil sebagai berikut:



Gambar 2 Google Translate

Jika diperhatikan dalam gambar diatas, Google Translate akan melakukan penerjemahan terhadap kalimat “Halo apa kabar” menjadi “Hello how are you” dalam bahasa Inggris. Jika dilakukan *highlighting* pada suatu kata dengan mengarahkan pointer mouse ke kata tertentu, dapat dilihat padanan kata tersebut dalam bahasa yang berlawanan. Misalkan dalam gambar berikut ini:



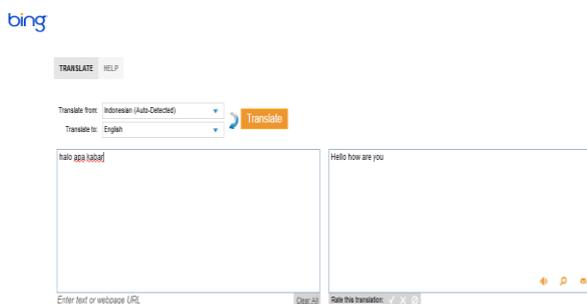
Gambar 3 Padanan Kata dalam Google Translate

Dapat dilihat dalam gambar diatas, kata “hello” dalam bahasa Inggris akan diberi tanda berpadanan dengan kata “halo” dalam bahasa Indonesia. Selain itu, jika kata “hello” tersebut diklik, akan muncul alternatif-alternatif terjemahan yang lainnya.

Dengan meninjau Google Translate ini, dapat disimpulkan bahwa secara umum aplikasi ini melakukan penerjemahan dengan melakukan *string matching* kata per kata. Memang sebenarnya ada proses lebih kompleks lainnya seperti peninjauan terhadap *idiom* ataupun pola kalimat lainnya, namun dalam makalah kali ini penulis akan mengabaikan hal tersebut.

Terdapat hal yang unik dalam contoh kasus diatas, yaitu penerjemahan kata “apa” dan “kabar” yang diterjemahkan menjadi “how are you”. Padahal, jika dilakukan penerjemahan secara kata per kata, terjemahan yang terjadi akan menjadi “what” dan “news”. Hal ini jugalah yang harus diperhatikan dalam aplikasi terjemah, yaitu kata-kata yang bersifat majemuk.

Hal yang sama juga terjadi ketika dilakukan uji coba terhadap aplikasi penerjemah yang lainnya, yaitu Bing Translator. Dengan menginputkan kata “halo apa kabar”, Bing Translator memberikan hasil sebagai berikut:



Gambar 4 Aplikasi Penerjemah Bing Translator

III. ALGORITMA KNUTH MORRIS PRATT

Algoritma Knuth Morris Pratt bekerja dengan memanfaatkan pergeseran yang semaksimal mungkin dalam pencocokan string dalam teks. Misalkan saja terdapat kata “kabar” yang ingin dicari padanannya dalam bahasa Inggris. Aplikasi penerjemah haruslah melakukan pencocokan kata untuk mencari padanan kata tersebut dalam bahasa asing. Haruslah diperhitungkan apakah kata tersebut merupakan bagian dari kata majemuk lainnya. Misalkan kata “kabar” yang dimaksud disini merupakan bagian dari kata “apa kabar” yang dapat dianggap sebagai satu kesatuan kata majemuk. Proses dengan menggunakan algoritma KMP akan menjadi sebagai berikut:

1. Sebelumnya dilakukan pencocokan dengan kata-kata lain yang terdapat dalam kamus, hingga akhirnya tiba dalam kalimat “apa kabar”, algoritma KMP akan mengenali terdapat kata “kabar” dalam kalimat tersebut dengan proses sebagai berikut:

Ket. Kata yang dibaris atas merupakan kata yang diperiksa, dan kata yang dibawah merupakan kata yang dicocokkan.

A	P	A		K	A	B	A	R
K	A	B	A	R				

Dalam proses pencocokan pertama tidak ditemukan kemiripan dalam kata tersebut, maka selanjutnya akan dilakukan penggeseran untuk pencocokan selanjutnya. Dalam kasus diatas, penggeseran akan dilakukan sejauh satu karakter, dan terus berulang selama tiga kali pencocokan selanjutnya (dalam tiga kali pencocokan selanjutnya tidak ditemukan kesamaan karakter), sehingga pencocokan selanjutnya adalah sebagai berikut:

A	P	A		K	A	B	A	R
				K	A	B	A	R

Dan akhirnya ditemukan bahwa kata “kabar” merupakan bagian dari kata “apa kabar”.

Dalam contoh lain dapat dilihat bahwa dalam algoritma KMP memiliki perhitungan khusus dalam melakukan pergeseran string. Misalkan untuk pencocokan kata “abcabd” dalam “abcabcabd”. Prosesnya adalah sebagai berikut:

A	B	C	A	B	C	A	B	D
A	B	C	A	B	D			

Dalam tahap awal dilakukan pencocokan mulai dari karakter paling kiri, dapat dilihat bahwa terdapat kecocokan hingga 5 karakter, namun pada karakter ke 6,

yaitu “D” terdapat perbedaan. Untuk menentukan sejauh apa pergeseran yang akan dilakukan, terlebih dahulu haruslah dilakukan penghitungan terhadap fungsi pinggiran untuk karakter yang cocok. Fungsi pinggiran merupakan ukuran awalan terpanjang dari suatu string P yang merupakan akhiran dari P[1..j].

Sebagai gambaran lengkap, berikut daftar fungsi pinggiran untuk kata “abcabd”:

A	B	C	A	B	D
0	0	0	1	2	0

Dalam kasus diatas, string yang cocok merupakan string “abcab” yang memiliki fungsi pinggiran ab (bernilai 2 karakter). Sehingga berdasarkan algoritma KMP, pergeseran yang dilakukan adalah sejauh panjang “abcab” dikurangi nilai fungsi pinggirannya yaitu 5-2=3. Maka pergeseran yang dilakukan adalah sejauh 3 karakter.

A	B	C	A	B	C	A	B	D
			A	B	C	A	B	D

Setelah dilakukan penggeseran akhirnya ditemukan kecocokan dalam string diatas.

IV. PENERAPAN ALGORITMA KMP DALAM APLIKASI PENERJEMAH

Untuk menguji cobakan algoritma KMP dalam aplikasi penerjemah, penulis melakukan percobaan dengan membuat suatu program dalam bahasa C++. Berikut beberapa pertimbangan dan analisa yang dilakukan dalam pembuatan program kali ini:

1. Aplikasi penerjemah akan menerima inputan kalimat-kalimat dalam bahasa Indonesia.
2. Selanjutnya akan dilakukan pemrosesan terhadap kalimat-kalimat tersebut menjadi kata-kata yang terpecah-pecah demi kemudahan dalam melakukan pencocokan string.
Misalkan untuk kata “halo apa kabar” akan dipecah sebagai berikut:

Halo
Apa
Kabar

Sebenarnya kata “apa” dan “kabar” haruslah dianggap sebagai satu kesatuan kata karena terjadi perbedaan makna jika kedua kata tersebut diterjemahkan secara terpisah, namun hal itu akan dilakukan pada tahapan yang selanjutnya.

3. Aplikasi penerjemah akan melakukan perbandingan kata dengan daftar kata-kata yang ada dalam kamus.
4. Perbandingan kata dilakukan berdasarkan kata dengan kemiripan yang paling tinggi, selain itu

perbandingan akan dilakukan dengan kata yang bersebelahan dengan kata tersebut, misalkan dalam kasus kata “apa kabar”, kata “apa” dan “kabar” sama-sama akan menemukan kecocokan dalam kata “how are you”, sehingga program mampu mengenali kata “apa kabar” sebagai satu kesatuan “how are you”. Untuk lebih jelasnya perhatikan dalam tabel berikut:

A	P	A		K	A	B	A	R
A	P	A						

Ket. Dalam kasus kata “apa”, ditemukan kecocokan dalam kata “apa kabar” yang diterjemahkan sebagai “how are you”.

A	P	A		K	A	B	A	R
				K	A	B	A	R

Ket. Dalam kasus kata “kabar”, kembali ditemukan kecocokan dalam kata “apa kabar” yang sama-sama diterjemahkan sebagai “how are you”.

Jika kasus seperti diatas terjadi, maka sistem akan mempertimbangkan kata “apa kabar” sebenarnya merupakan satu kesatuan yang bermakna “how are you”. Berbeda jika kata yang diinputkan adalah “apa itu”, tidak akan dianggap sebagai satu kesatuan dengan “how are you”.

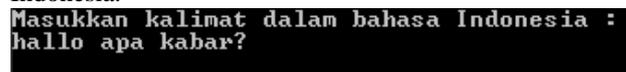
Langkah inilah yang dilakukan dalam uji coba kali ini untuk menemukan padanan kata yang paling sesuai untuk kata-kata tersebut juga untuk kata majemuk. Pencocokan ini dilakukan dengan algoritma KMP.

5. Selanjutnya program akan menampilkan hasil terjemahan dalam bahasa Inggris sesuai dengan pencocokan kata yang dilakukan.

V. UJI COBA PROGRAM

Setelah dilakukan pembuatan program dalam bahasa C++ selanjutnya dilakukan uji coba program untuk menguji keberhasilan program ini. Berikut adalah hasil uji coba tersebut:

1. Sistem menerima inputan kalimat dalam bahasa Indonesia.



Gambar 5 Inputan Awal

2. Sistem memecah kalimat menjadi kata-kata yang terpisah.

```
Kalimat tersebut dipecah menjadi:
hallo
apa
kabar
```

Gambar 6 Pemecahan Kata

- Sistem mulai menerjemahkan kata satu per satu.

```
Penerjemahan Kata
Kata : hallo
Arti : hello

Kata : apa
Arti : what -- pertimbangan arti 'how are you' (kemiripan 3/8)

Kata : kabar
Arti : news -- pertimbangan arti 'how are you' (kemiripan 5/8)

Terjemahan : hello what news
Terjemahan alternatif : hello how are you
```

Gambar 7 Hasil Terjemahan

Dapat dilihat dalam *screenshot* program diatas, sistem melakukan penerjemahan kata per kata dengan turut mempertimbangkan kehadiran kata-kata yang mungkin sepadan, seperti kata “apa kabar”. Dalam percobaan kali ini, sistem melakukan penerjemahan ke dalam dua kalimat yang berbeda dengan mempertimbangkan seperti penjelasan yang dilakukan pada bagian IV sebelumnya.

Selanjutnya juga dilakukan uji coba untuk kata majemuk lainnya, misalnya “anak tiri”, dengan hasil sebagai berikut:

- Sistem menerima inputan kalimat dalam bahasa Indonesia.

```
Masukkan kalimat dalam bahasa Indonesia :
anak tiri
```

Gambar 8 Inputan Awal Versi 2

- Sistem melakukan pemecahan kata per kata.

```
Kalimat tersebut dipecah menjadi:
anak
tiri
```

Gambar 9 Pemecahan Kata Versi 2

- Sistem mulai menerjemahkan kata satu per satu.

```
Penerjemahan Kata
Kata : anak
Arti : child -- pertimbangan arti 'stepchild' (kemiripan 4/4)

Kata : tiri
Arti : step -- pertimbangan arti 'stepchild' (kemiripan 4/4)

Terjemahan : child step
Terjemahan alternatif : stepchild
```

Gambar 10 Hasil Terjemahan Versi 2

Dapat dilihat dari hasil diatas, pada awalnya program menganggap kata “anak tiri” sebagai dua buah kata yang terpisah, yaitu “anak” dan “tiri”. Namun dalam pemrosesannya, akhirnya program menyadari bahwa sebenarnya “anak tiri” merupakan kata majemuk yang

berarti “stepchild”.

Jika diperhatikan dalam hasil uji diatas terdapat keterangan kemiripan 4/8, maksudnya adalah sebagai berikut:

Dalam pemrosesan yang pertama, yaitu kata “anak” dilakukan sebagai berikut:

A	N	A	K	
C	H	I	L	D

Kemiripan : 4/4

A	N	A	K		T	I	R	I
S	T	E	P	C	H	I	L	D

Kemiripan: 4/8

Karena kemiripan “anak” dibandingkan “anak tiri” lebih tinggi, maka pada tahap awal kata yang dipilih adalah kata “anak” yang diartikan “child”. Namun ketika sampai pada kata tiri:

T	I	R	I
S	T	E	P

Kemiripan: 4/4

A	N	A	K		T	I	R	I
S	T	E	P	C	H	I	L	D

Kemiripan: 4/4

Berdasarkan kemiripan memang pada awalnya dipilih kata “step” namun akhirnya program mempertimbangkan bahwa sebenarnya kata “anak tiri” merupakan satu kesatuan yang berarti “stepchild” karena itulah pada akhir program muncul dua arti yang salah satunya adalah kata “stepchild”.

Dari hasil dua uji coba diatas, dapat disimpulkan bahwa program telah berhasil diimplementasikan dengan baik.

Sebagai keterangan tambahan, program membaca arti kata dengan berdasarkan pada file eksternal yang berisi kata dalam bahasa indonesia disertai terjemahannya dalam bahasa Inggris, misalnya sebagai berikut:

```
apel=apple#
buku=book#
apa kabar=how are you#
play=bermain#
```

Kata yang berada disebelah kiri merupakan kata dalam bahasa Indonesia, sementara yang disebelah kanan merupakan kata dalam bahasa Inggris (dipisahkan oleh tanda sama dengan “=”). Setiap kata yang berbeda dipisahkan juga oleh tanda pagar “#”.

VI. KESIMPULAN

Setelah dilakukan uji coba terhadap implementasi algoritma KMP dalam aplikasi penerjemah kata, dapat disimpulkan bahwa algoritma ini memang dapat digunakan untuk melakukan penerjemahan kata dengan berdasarkan pada *string matching* atau pencocokan string.

Diperlukan pengembangan lebih lanjut untuk program ini, terutama dalam masalah kemangkusan algoritma yang digunakan.

REFERENSI

- [1] Munir, Rinaldi, *Diktat Kuliah IF3051 Strategi Algoritma*. Program Studi Teknik Informatika ITB, 2012.
- [2] Munir, Rinaldi, *Matematika Diskrit*. Informatika Bandung, Bandung, September 2010.

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, 16 Desember 2012



Okharyadi Saputra
13510072