

# Algoritma *Predictive Text Input* Sederhana Berbasis Algoritma *Depth First Search*

Ardian Franindo<sup>1</sup>, Tessa Ramsky<sup>2</sup>, Jamal Saepul Aziz<sup>3</sup>

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung

E-mail : [if14106@students.if.itb.ac.id](mailto:if14106@students.if.itb.ac.id)<sup>1</sup>,  
[if14124@students.if.itb.ac.id](mailto:if14124@students.if.itb.ac.id)<sup>2</sup>, [if14156@students.if.itb.ac.id](mailto:if14156@students.if.itb.ac.id)<sup>3</sup>

## Abstrak

Layanan Pesanan Singkat atau *Short Message Services* (SMS) adalah layanan seluler paling populer. Karena keterbatasan perangkat bergerak yang digunakan, pengguna membutuhkan kombinasi tombol yang lebih banyak untuk mengirim pesan dibanding menggunakan perangkat seperti komputer yang menggunakan papan ketik biasa. *Predictive Text Input* adalah salah satu metode untuk mereduksi penggunaan tombol dalam mengirim pesan SMS. *Predictive Text Input* memperkirakan kata yang akan dituliskan pengguna berdasarkan masukan tombol dari pengguna dan daftar kata-kata yang sering dipergunakan. Algoritma *Predictive Text Input* sederhana dapat dibuat dengan berbasiskan struktur data pohon dan salah satu algoritma penelusuran pohon, dalam makalah ini adalah algoritma *depth first search*. Kata-kata yang sering dipergunakan disusun sedemikian rupa dalam struktur pohon sehingga memudahkan dalam pemrosesan dengan algoritma *depth first search*.

**Kata kunci:** *short message service, predictive text input, pohon, depth first search*

## 1. Pendahuluan

Layanan Pesan Singkat atau *Short Message Service* (SMS) adalah suatu layanan pengiriman pesan lewat jaringan seluler. Dibandingkan dengan layanan lain yang disediakan penyedia jasa seluler, SMS adalah yang paling populer. Menurut GSM World, sekitar 1 trilyun SMS dikirim pada tahun 2005 kemarin.

Meskipun SMS sangat populer, sebenarnya terdapat beberapa keterbatasan dalam penggunaan SMS ini. Salah satu keterbatasan yang ada muncul karena kebanyakan perangkat bergerak menggunakan *keypad* ITU yang hanya terdiri dari tombol angka 0 sampai dengan tombol angka 9 ditambah tombol '\*' dan tombol '#'. Hal ini mengakibatkan kombinasi tombol yang diperlukan untuk menulis sebuah kata lebih banyak dibandingkan dengan menggunakan papan ketik (*keyboard*) biasa.

Pada perangkat bergerak modern, kemampuan pengiriman pesan biasanya dilengkapi dengan kemampuan untuk memperkirakan kata yang ingin dimasukkan pengguna. Kemampuan ini disebut *predictive text input* atau perkiraan teks masukan. Salah satu algoritma yang populer adalah T9 dari Tegic Communication.

## 2. Rumusan Masalah

Makalah ini terutama ditujukan untuk membahas algoritma sederhana tetapi mangkus untuk memperkirakan kata yang ingin dimasukkan pengguna. Struktur data yang digunakan adalah pohon untuk menyimpan semua kata yang mungkin akan digunakan pengguna. Sementara algoritma untuk menghasilkan kata dari masukan pengguna ialah algoritma pencarian mendalam atau *depth first search* (DFS).

## 3. Pohon Kata

Untuk memudahkan dalam perkiraan kata, maka kata-kata yang digunakan disusun dalam bentuk pohon. Urutan penyusunan kata pada pohon berdasarkan pada kelompok masukan pada *keypad* ITU, panjang kata dan kekerapan penggunaan. Kata-kata yang karakter-karakternya berasal dari *keypad* yang sama dikelompokkan dalam satu anak pohon. Kata dengan jumlah karakter lebih sedikit ditempatkan pada level yang lebih rendah, dan kata dengan kekerapan penggunaan yang lebih tinggi ditempatkan pada cabang yang lebih kiri. Akar bagi setiap anak pohon selain pada level nol adalah simpul dengan kekerapan yang paling tinggi.

Berikut ini adalah contoh pohon dari sebuah tabel kata yang sudah terurut sesuai kriteria penyusunan kata pada pohon kata.

Tabel kata dengan panjang satu karakter

Tombol	Karakter	Kekerapan
2	a	50%
	b	30%
	c	20%
3	d	35%
	e	50%
4	f	15%
	g	20%
	h	25%
5	i	55%
	j	30%
	k	40%
6	l	30%
	m	30%
	n	35%
7	o	35%
	p	30%
	q	5%
	r	35%
8	s	30%
	t	40%
	u	50%
9	v	10%
	w	15%
	x	5%
	y	70%
	z	10%

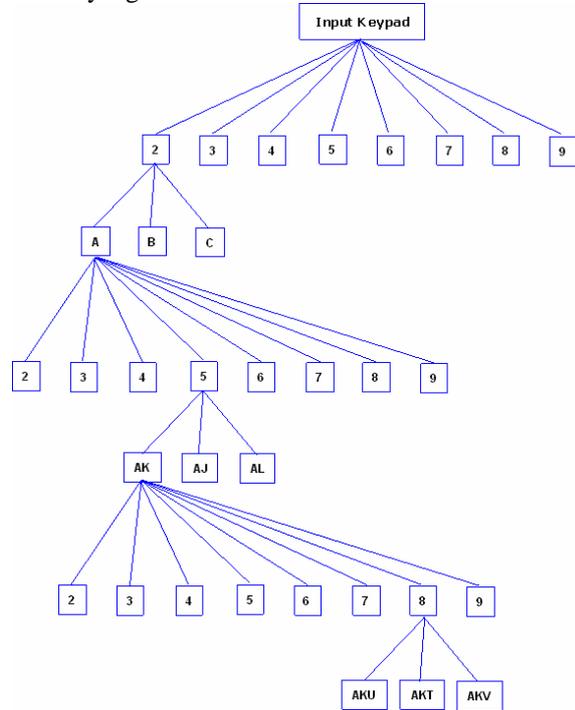
Tabel kata dengan panjang dua karakter

Tombol 1	Tombol 2	Kata yang terbentuk
2	5	Ak
		Aj
		Al
		Bk
		Bj
		Bl
		Cl

Tabel kata dengan panjang tiga karakter

Tombol 1	Tombol 2	Tombol 3	Kata yang terbentuk
2	5	8	Aku
			Akt
			Aju
			Alu
			Alv
			Blu
			Blt
			Clu

Pohon yang terbentuk



#### 4. Algoritma Perkiraan Kata Masukan

##### 4.1. Prinsip Perkiraan Kata Masukan

Tinjau kembali pohon yang terdapat pada bagian 3. Pertama-tama penelusuran dimulai pada simpul akar yang melambangkan *string* kosong (belum ada masukan). Ketika pengguna memberikan masukan, maka penelusuran akan berlanjut pada simpul yang bersesuaian dengan masukan tersebut, dan langsung menuju cabang paling kiri dari simpul masukan tersebut. Penelusuran akan terus berlanjut sampai salah satu dari beberapa kondisi berikut terpenuhi :

1. Pengguna menekan tombol 0, 1, dan '#', maka program berhenti
2. Pengguna menekan tombol '\*', artinya pengguna menginginkan solusi lain dari solusi yang diberikan. Penelusuran dilanjutkan dengan melakukan runut-balik dan menampilkan isi daun solusi dari daun di sebelah kanan daun solusi semula.

##### 4.2. Kode-Semu Perkiraan Kata Masukan

Kode-semu (*pseudocode*) dari algoritma perkiraan kata masukan adalah :

```

function Cabang(input v : String, input
i : integer)-> String
{

```

Menghasilkan cabang ke-*i* dari daun *v*. Penomoran cabang dimulai dari kiri ke kanan. Cabang paling kiri bernomor 1.

Masukan : *v* : Daun berupa string  
*i* : Nomor cabang yang dicari

Keluaran : Cabng nomor ke-*I* dari *v*

```
}  
  
function NextDaun(input v : String)->  
String  
{  
    Menghasilkan daun di sebelah kiri  
    cabang v. Artinya NextDaun(v) akan  
    menghasilkan saudara dari v yang  
    terletak di sebelah kirinya  
  
    Masukan : Daun berupa string v  
    Keluaran : Daun yang terletak di  
    sebelah kanan daun v, atau NULL bila  
    tidak ada daun di sebelah kanan v  
}
```

```
procedure Perkiraan_Kata_Masukan(input  
v : String  
)  
{  
    Memperkirakan kata yang ingin  
    dimasukkan pengguna berdasarkan  
    masukan tombol yang diberikan.  
  
    Masukan : v adalah simpul awal  
    kunjungan  
}
```

**Deklarasi**  
*i* : integer

#### Algoritma

```
if (v != NULL) then  
    if level(v) mod 2 = 1  
        v <- Cabang(v, 1)  
    write(v)  
    MASUKAN_PENGGUNA  
    Select_case  
    2 : Perkiraan_Kata_Masukan(Cabang(v,  
        2))  
    3 : Perkiraan_Kata_Masukan(Cabang(v,  
        3))  
    4 : Perkiraan_Kata_Masukan(Cabang(v,  
        1))  
    5 : Perkiraan_Kata_Masukan(Cabang(v,  
        5))  
    6 : Perkiraan_Kata_Masukan(Cabang(v,  
        6))  
    7 : Perkiraan_Kata_Masukan(Cabang(v,  
        7))  
    8 : Perkiraan_Kata_Masukan(Cabang(v,  
        8))  
    '*' :  
        Perkiraan_Kata_Masukan(NextDaun(  
            v))  
    else : STOP
```

## 5. Kesimpulan dan Saran

Algoritma *depth first search* pada pohon kata yang terurut menurut panjang kata dan kekerapan penggunaannya dapat digunakan sebagai algoritma pada *Predictive Text Input*. Penggunaan algoritma ini dapat mempersingkat waktu penulisan pesan pada SMS. Meskipun algoritma ini tidak secanggih algoritma T9 buatan Tegic Communication, namun dapat digunakan sebagai alternatif yang cukup mangkus.

### Daftar Pustaka

1. H. Komatsu, S. Takabayashi, T. Masui, *Corpus-Based Predictive Text Input*, National Institute of Advanced Industrial Science and Technology, 2005
2. Rinaldi Munir, *Diktat Kuliah IF2251 Strategi Algoritmik*, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. Bandung, 2006
3. [www.gsmworld.com/](http://www.gsmworld.com/) diakses tanggal 19 Mei 2006 pukul 10.00