BAB 2 LANDASAN TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi landasan untuk penelitian model *phonetic similarity* bahasa Indonesia.

2.1. Phonetic similarity

Phonetic similarity secara sederhana dapat didefinisikan sebagai studi mengenai kemiripan antar bunyi. Dengan mengembangkan model *phonetic similarity* diharapkan terciptanya suatu metrik untuk mengukur kemiripan antar bunyi.

2.1.1 Fonologi

Fonologi merupakan sistematis cara penggunaan suara dalam kapasitasnya sebagai penyalur ataupun pembeda makna dalam bahasa (manusia). Dalam linguistik, fonologi merupakan subbidang yang berurusan dengan sistem suara dalam bahasa.

2.1.2. Fonem

Fonem merupakan salah satu subjek analisis linguistik dalam fonologi. Fonem didefinisikan sebagai satuan unit bunyi terkecil dalam kapasitasnya sebagai pembeda makna. Pada penelitian ini, fonem merupakan satuan unit bunyi yang akan digunakan untuk meneliti tentang kemiripan antar bunyi.

2.1.3. Penerapan phonetic similarity

Terdapat berbagai bidang yang memanfaatkan bidang *phonetic similarity*. Bidang-bidang tersebut diantaranya:

1. *Speech recognition*, *phonetic similarity* dapat berperan sebagai salah satu komponen dalam suatu sistem *speech recognition*. Misalnya pada aplikasi *speech-to-text* dimana program menerima masukan bunyi dan mengeluarkan hasil berupa tulisan. Pada sistem demikian yang berlandaskan kamus pemetaan antara bunyi dan tulisan, ada saatnya bunyi yang dipersepsi oleh mesin kurang akurat sehingga kata yang terdapat pada kamus pemetaan gagal dipetakan. Dengan adanya *phonetic similarity*, bunyi yang tidak sama persis dengan yang ada di kamus tetap dapat dipetakan.

2. *Pun jokes generator*, merupakan sistm pembangkit lelucon berbasis plesetan. Idenya ialah bahwa plesetan biasanya merupakan pasangan kata yang bunyinya mirip.

3. *Cognates*, merupakan pasangan kata yang memiliki asal etimologi yang sama. Biasanya kata tersebut berasal dari bahasa yang berbeda. Misalnya *night* pada bahasa Inggris dan *nuit* pada bahasa Perancis. Terdapat kecenderungan bahwa *cognates* memiliki kemiripan bunyi. oleh karena itu, *phonetic similarity* dapat digunakan sebagai salah satu alat untuk mencoba mendeteksi *cognates*.

2.1.4. Model phonetic similarity Stephen Rooney

Penelitian sebelumnya mengenai phonetic similarity dilakukan oleh Stephen Rooney [STR05]. Model phonetic similarity yang dikembangkan menekankan pada hubungan kemiripan antar fonem. Set fonem yang digunakan ialah set fonem untuk bahasa Inggris yang terdapat pada situs CMU talking dictionary. Terdapat 39 fonem pada set fonem tersebut. Tujuan dari penelitian ialah mendapatkan hubungan kemiripan untuk setiap pasangan fonem sebagai alat bantu untuk menghitung kemiripan bunyi antar kata. Untuk mendapat hubungan kemiripan tersebut dilakukan metode empiris dengan metode kuesioner. Kuesioner disebarkan secara online dan memiliki format pertanyaan sebagai berikut:

Nilai kemiripan antara pasangan bunyi berikut o – contoh: odd dan @ - contoh: at

Pada halaman pengerjaan terdapat link untuk mendengarkan bunyi dari fonem yang ditanyakan serta contoh bunyi tersebut pada kata bahasa Inggris yang umum digunakan. Pada halaman pengerjan juga terdapat checkbox dengan rentang nilai 0 sampai 10 untuk menilai kemiripan antara pasangan fonem yang diberikan.

Pada akhir penelitian diperoleh matriks 39 x 39 yang berisi nilai cost antara tiap pasangan fonem. Dengan bermodalkan matriks tersebut, penghitungan cost antar kata yang memiliki jumlah fonem yang sama dapat dilakukan. Sebagai contoh,

untuk kata *chime* dan *time* penghitungan *cost* dilakukan dengan langkah-langkah sebagai berikut:

1. Konversi kedua kata tersebut menjadi simbol fonem menggunakan CMU talking dictionary, sehingga diperoleh CH AY M sebagai simbol fonem dari chime dan T AY M sebagai simbol fonem dari time.

2. Nilai cost ditentukan oleh nilai pada matriks untuk sel (CH,T),(AY,AY), dan (M,M). Nilai pada sel-sel tersebut dijumlahkan lalu dibagi dengan panjang fonem salah satu kata. Nilai cost ialah (10+0+0)/3 = 3,33.

2.1.5. Model phonetic similarity STANDUP

Model phonetic similarity STANDUP [RGH08] menggunakan simbol unisyn sebagai referensi fonem [SUS]. Model ini melibatkan penggunaan atribut untuk menghitung cost substitusi antar fonem. Setiap atribut memiliki set nilai masing-masing. Atribut-atribut tersebut tersebar dalam tiga level. Level 1 memiliki satu atribut VC, yang membedakan antara fonem vokal dan konsonan. Level 2 terdapat 6 atribut: height, frontness, rounding untuk fonem vokal (tabel 2.1), dan voicing, place, manner untuk fonem konsonan (tabel 2.2).

Terdapat juga atribut pada level 3. Untuk tiap triplet nilai atribut level 2 terdapat satu kelas kecil, tapi kelas kecil tersebut mungkin tidak terdiri dari satu anggota. Misalnya pada fonem vokal, triplet M-C-U merupakan kelas dari {@,@r, @@r}. Untuk tiap triplet tersebut terdapat satu atribut level 3. Atribut tersebut memiliki kemungkinan satu nilai untuk tiap fonem.

Sehingga, setiap fonem memiliki satu nilai atribut level 1, tiga nilai atribut level 2, dan satu nilai atribut level 3. Setiap atribut memiliki nilai cost. Penghitungan nilai cost antar fonem dilakukan berdasar nilai cost pada atribut tersebut. Dua fonem diberi nilai cost pada level tertinggi dimana mereka tidak memiliki nilai atribut yang sama. Artinya, jika fonem F_1 dan fonem F_2 memiliki nilai atribut level 1 yang sama, maka yang dipertimbangkan ialah nilai atribut level 2 mereka, dan seterusnya. Pada level tersebut, nilai cost pasangan fonem ialah jumlah cost pada atribut yang memiliki nilai

Tabel 2. 1 Atribut level 2 untuk fonem vokal

Symbol	H	F	R	Symbol	H	F	R	Symbol	H	F	R
@	MC	C	U	eir	MC	F	U	or	M	B	R
@@r	MC	C	U	er	MO	F	U	ou	MC	B	R
@r	MC	C	U	i	MCC	F	U	our	MC	B	R
a	O	F	U	ii	C	F	U	ow	OC	B	U
ae	O	F	N	ii;	C	F	U	owr	OC	B	U
aer	O	F	N	ir	C	F	U	uh	MO	B	U
ai	O	F	N	ir;	C	F	U	ur	C	C	R
ar	O	F	U	oi	MO	B	R	ur;	C	C	R
e	MO	F	U	oir	MO	B	R	uu	C	C	R
ei	MC	F	U	oo	M	B	R	uu;	C	C	R

Tabel 2. 2 Atribut level 2 untuk fonem konsonan

Symbol	V	P	M	Symbol	V	P	M	Symbol	V	P	M
?	U	G	SP	l	U	L	F	sh	U	PA	F
b	V	BL	SP	l!	V	A	FL	t	U	A	SP
ch	U	PA	A	m	V	BL	N	t^	V	PA	FL
d	V	A	SP	m!	V	LD	N	th	U	D	F
dh	V	D	F	n	V	A	N	v	V	LD	F
f	U	LD	F	n!	V	P	N	w	V	LV	A
g	V	V	SP	ng	V	V	N	x	V	V	F
h	U	G	F	p	U	BL	SP	y	V	P	A
hw	U	LV	A	r	V	PA	F	z	V	A	F
jh	V	PA	A	s	U	A	F	zh	V	PA	F
k	U	V	SP								

berbeda. Nilai cost atribut VC pada level 1 ialah 1 (nilai maksimum). Nilai cost pada atribut level 2 ialah 0,28 untuk fonem konsonan dan 0,15 untuk fonem vokal. Nilai cost pada atribut level 3 ialah 0,15.

Sebagai contoh, fonem f dan v memiliki atribut level 1 yang sama (keduanya merupakan fonem konsonan), tapi nilai atribut level 2 mereka secara berturut-turut ialah voicing=U, placing=LD, manner=F dan voicing=V, placing=LD, manner=F, perbedaan satu nilai atribut. Karena nilai cost pada atribut level 2 bernilai 0,28, maka nilai cost pasangan f dan v ialah 0,28. Sebaliknya fonem k memiliki nilai atribut level 2 voicing=U , place=V , manner=SP memiliki perbedaan dalam tiga atribut level 2 dengan fonem v, sehingga cost pasangan k dan v ialah 0,84.

Untuk menilai kemiripan antar kata, diterapkan algoritma Levenshtein distance yang dimodifikasi. Dilakukan normalisasi terhadap panjang kata dan nilai operasi substitusi sesuai dengan nilai kemiripan bagi pasangan yang disubstitusikan.

2.2. Kamus fonetik Amalia Zahra

Pada penelitian yang dilakukan oleh Amalia Zahra [ZAR08], dikembangkan sistem pengenalan suara untuk bahasa Indonesia. Sebagai salah satu penunjang sistem tersebut, dikembangkan kamus fonetik bahasa Indonesia. Kamus fonetik berisi daftar simbol untuk merepresentasikan bunyi dalam suatu bahasa. Untuk menyusun kamus fonetik bahasa Indonesia, dibutuhkan pengetahuan yang cukup mengenai fonologi bahasa Indonesia.

Metode yang digunakan untuk memperoleh kamus fonetik yang paling sesuai dengan bahasa Indonesia ialah melakukan pengujian dengan menggunakan berbagai kamus fonetik, termasuk kamus fonetik Amalia Zahra. Kamus fonetik Amalia Zahra diperoleh dari pengamatan terhadap hasil eksperimen menggunakan kelima kamus fonetik lainnya dan pengetahuan mengenai fonologi bahasa Indonesia.

Tabel 2. 3 Kamus fonetik Amalia Zahra

Simbol Fonetik	Contoh Bunyi
p	pasar, asap
b	batu, sabun
t	tari, padat
d	dari, adat
c	cari, acar
j	jari, ujar
k	kabar, dekat
g	gaji, tugas
f	sifat, vital
s	satu, besar
z	ziarah, zakat
sy	syarat, marsya
kh	akhir, khusus
h	hati, pahat
m	mata, amal
n	nasi, makan

Simbol Fonetik	Contoh Bunyi
ng	mengapa, angan
ny	banyak, nyaring
l	lari, batal
r	raja, lapar
w	wanita, kawan
y	saya, yakin
kk	tidak, bakwan
a	anda, api
i	nila, ini
u	Kamu, buta
e	sate, monyet
ax	Benar, belum
o	Bakso, tokoh
ay	andai, balai
oy	amboy, sepoy
aw	atau, kerbau
ey	mei

Hasil yang diperoleh pada eksperimen cukup memuaskan, sehingga dapat dikatakan bahwa kamus fonetik Amalia Zahra sesuai untuk digunakan pada bahasa Indonesia.

2.3. Levenshtein Distance

Levenshtein distance merupakan metrik yang digunakan untuk mengetahui jarak antara dua *sequence*. Levenshtein distance merupakan salah satu bentuk pengkhususan algoritma edit distance. Levenshtein Distance sering digunakan terutama dalam konteks string. Namun dinilai cukup tepat untuk juga diterapkan dalam konteks phonetic.

Jarak antara dua kata antara dua string ditentukan oleh jumlah minimal operasi yang dibutuhkan untuk merubah satu string ke string lainnya. Operasi yang dapat dilakukan ialah operasi substitusi, *insertion*, dan *deletion.*

Sebagai contoh, jarak antara string Peter dan Petrelli ialah 4. Dibutuhkan minimal empat operasi untuk merubah string Peter menjadi Petrelli atau sebaliknya:

- Peter → Petel (substitusi r dengan l)
- Petel → Petrel (insert r)
- Petrel → Petrell (insert l)
- Petrell → Petrelli (insert i)

Levenshtein distance dihitung menggunakan metode dynamic programming. Dynamic programming merupakan metode untuk menemukan solusi masalah yang memiliki karakteristik *overlapping subproblems* dan *optimal substructure*. *Overlapping subproblems* artinya masalah yang dapat dipecah menjadi submasalah yang dapat digunakan kembali. *Optimal substructure* berarti solusi global optimal dapat diperoleh dari solusi submasalah optimal. Oleh karena adanya karakteristik tersebut, secara tipikal metode dynamic programming menggunakan tabel atau matriks dalam pencarian solusi. Gambar 2.1 merupakan pseudocode dari Levenshtein distance. Untuk Levenshtein distance, nilai untuk operasi insertion, deletion, maupun substitution masing-masing bernilai 1. Tabel 2.4 merupakan contoh matriks yang dihasilkan untuk string “peter” dan “petrelli”.

```
function LEVENSHTEIN-DISTANCE (string1, string2)

n ← LENGTH(string1)

rn ← LENGTH(string2)

Create a distance matrix distance[n+l,m+l]

distance[O, O] ← 0

for each column i from 0 to n do

    for each row j from 0 to rn do

        distance[i,j] q-- MIN( distance[i- 1,j] + ins-cost(string1_j),

                               distance[i- 1,j- 1] + subst-cost(string2_j, string1_i),

                               distance[ i,j- 1] + ins-cost(string2_j))

return distance[n,m]
```

Gambar 2. 1 *Pseudo-code* Levenshtein distance

Tabel 2. 4 Contoh matriks hasil pengerjaan Levenshtein distance

		p	**e**	**t**	**e**	**r**
	0	1	2	3	4	5
p	1	0	1	2	3	4
e	2	1	0	1	2	3
t	3	2	1	0	1	2
r	4	3	2	1	1	1
e	5	4	3	2	1	2
l	6	5	4	3	2	2
l	7	6	5	4	3	3
i	8	7	6	5	5	4