



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

2.1 Terjemahan

Menurut Kamus Besar Bahasa Indonesia (Departemen Pendidikan Nasional, 2007), terjemahan adalah salinan bahasa atau pengalihan bahasa dari suatu bahasa ke bahasa lain. Pengalihan bahasa dapat dilakukan pada kata atau kumpulan kata agar menjadi bahasa lain. Terjemahan berasal dari kata dasar terjemah dengan tambahan imbuhan –an. Sedangkan penerjemahan berarti proses, cara, perbuatan menerjemahkan atau pengalihbahasaan. Orang yang berprofesi sebagai pengalih bahasa disebut dengan penerjemah. Seorang penerjemah mempunyai penghasilan yang cukup tinggi. Ada yang mengupahnya per hari, per jam, bahkan per sekian menit. Contoh terjemahan, yaitu terjemahan dari Bahasa Indonesia ke dalam Bahasa Mandarin dan dari Bahasa Mandarin ke dalam Bahasa Indonesia. Aktivitas menerjemahkan dapat dilakukan dengan mengakses kamus atau menggunakan jasa penerjemah. Contoh hasil terjemahan yang banyak dijumpai sehari-hari, yaitu novel-novel luar negeri yang diterjemahkan ke dalam Bahasa Indonesia.

2.2 Bahasa Indonesia

Bahasa Indonesia merupakan bahasa resmi yang digunakan Bangsa Indonesia. Bahasa Indonesia saat ini sudah menganut ejaan resmi yang pemakaiannya telah digunakan mulai 16 Agustus 1972. Indonesia adalah salah satu negara yang memiliki populasi penduduk terpadat di dunia. Karena kepadatan penduduknya, penduduk Indonesia berhasil menjadikan Bahasa

Indonesia menjadi salah satu bahasa yang paling banyak digunakan (Sheddon, 2004).

Aras kesadaran para pemuda Indonesia yang tergabung dalam perkumpulan pergerakan mengangkat Bahasa Melayu menjadi Bahasa Indonesia. Ini yang menjadikan Bahasa Indonesia menjadi bahasa persatuan untuk seluruh Bangsa Indonesia. Hari dimana Bahasa Indonesia dikenal sebagai bahasa persatuan diperingati sebagai hari Sumpah Pemuda yang jatuh pada tanggal 28 Oktober 1928.

Bahasa Indonesia mempunyai aturan tersendiri dalam penyusunan pola tata bahasa. Adapun setiap kata dalam Bahasa Indonesia dibedakan menjadi beberapa jenis yang disebut tata bahasa baku (Alwi, 2003). Jenis kata menurut tata bahasa baku dijelaskan secara singkat sebagai berikut.

1. Verba dari segi bentuk terbagi atas verba asal dan verba turunan, sedangkan verba dari segi perilaku sintaksisnya terbagi atas verba transitif, verba tak transitif, dan verba berpreposisi.
2. Adjektiva dari segi bentuknya terbagi atas adjektiva dasar dan adjektiva turunan, sedangkan adjektiva dari segi perilaku sintaksisnya terdiri atas adjektiva atributif, predikatif, dan adverbial atau keterangan.
3. Adverbial dari segi bentuknya terbagi atas adverbial tunggal dan adverbial gabungan.
4. Nomina dari segi bentuknya dapat dibedakan menjadi nomina dasar, turunan, afiks, dan morfonomik afiks nomina.
5. Pronomina penunjuk terdiri atas pronomina penunjuk umum, penunjuk tempat, dan penanya.

6. Numeralia dibagi atas numeralia pokok, tingkat, dan pecahan.
7. Kata tugas diklasifikasikan menjadi preposisi, konjungtor, interjeksi, artikula dan partikel penegas.

2.3 Bahasa Mandarin

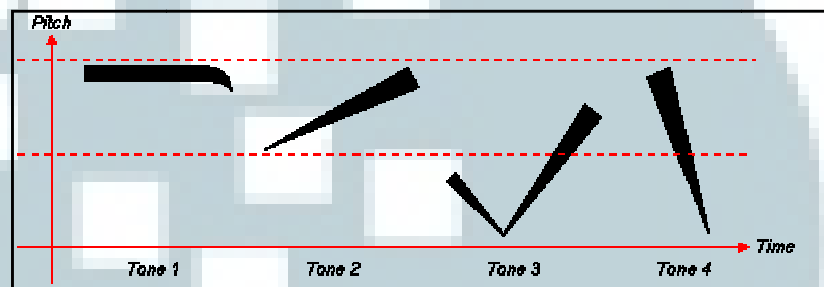
Bahasa Mandarin atau bahasa Tionghoa (汉语, 华语, atau 中文; *hànyǔ*, *huáyǔ*, atau *zhōngwén*) merupakan bahasa asal penduduk asli Cina. Bahasa Mandarin merupakan bahasa universal nomor dua setelah Bahasa Inggris. Menurut Kane (2006), bahasa ini terdiri dari dua macam karakter, yaitu *Simplified Chinese Character* (简体字; *Pinyin: Jiǎntǐzì*) dan *Traditional Chinese Character* (繁体字; *Pinyin: Fántǐzì*)

Traditional Chinese Character digunakan di Taiwan dan Hongkong, sedangkan *Simplified Chinese Character* lebih banyak digunakan di daratan Cina. Pengucapan bahasa ini menggunakan fonologi dengan aturan-aturan yang sudah ditetapkan yang disebut dengan *pinyin*. Setiap tekanan nada dalam pengucapan mengandung sebuah arti. Jika terjadi salah pengucapan tekanan nada, maka artinya juga akan berbeda. Sebuah kata yang dideskripsikan dalam *pinyin* bisa memiliki satu atau lebih arti. Perbedaan arti disesuaikan dengan dimana sebuah aksara mandarin ditempatkan. Contohnya adalah kata 学 (*pinyin: xué*). *Xué* ketika berdiri sendiri berarti belajar, tetapi ketika digabungkan dengan 校 (*pinyin: xiào*) dan menjadi 学校 (*pinyin: xuéxiào*) artinya menjadi sekolah.

initials	finals	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
a	o	a	o	e	ai	ei	ao	ou	an	en	ang	eng	ong	u	uo	uai	ui	uan	uang	un	ueng	i	ia	ie	iao	iu	ian	iang	in	ing	long	ü	üe	üan	ün		
b	bo	ba	bo	bei	bai	bei	bao		ban	ben	bang	beng	bu									bi	bie	biao		bian	bin	bing									
p	po	pa	po	pei	pai	pei	pao	pou	pan	pen	pang	peng	pu									pi	pie	piao		pian	pin	ping									
m	mo	ma	mo	mai	mai	mai	mao	mou	man	men	man	meng	mu									mi	mie	miao		miu	min	ming									
f	fo	fa	fo		fei	fei		fou	fan	fen	fang	feng	fu																								
d	de	da		dai	dai	dai	dao	dou	dan	den	dang	deng	du	duo				dui	duan	dun		di	dai	dai	diao	diao	dian	dian	ding								
t	te	ta		tai	tai	tai	tao	tou	tan		tang	teng	tu	tuo				tui	tuan	tun		ti	tie	tiao		tian											
n	ne	na		nai	nai	nai	nao	nou	nan	nen	nang	neng	nu	nuo					nuan	nun		ni	nie	niao		niu	niang	ning			nü	nüe					
l	le	la		lai	lai	lai	lao	lou	lan		lang	leng	lu	luo					luan	lun		li	lie	liao		liu	lian	ling		lü	lüe						
g	ge	ga		gai	gai	gai	gao	gou	gan	gen	gang	geng	gu	guo	guai			gui	guan	gun																	
k	ke	ka		kai	kai	kai	kao	kou	kan	ken	kang	keng	ku	kua	kuo	kuai		kui	kuan	kun																	
h	he	ha		hai	hai	hai	hao	hou	han	hen	hang	heng	hu	hua	huo	huai		hui	huan	hun																	
z	zi	za		zai	zai	zai	zao	zou	zan	zen	zang	zeng	zu	zuo				zui	zuan	zun		zi															
c	ci	ca		cai	cai	cai	cao	cou	can	cen	cang	cheng	cu	cuo				cui	cuan	cun		cin															
s	si	sa		sai	sai	sai	sao	sou	san	sen	sang	seng	su	suo				sui	suan	sun		si															
zh	zhi	zha		zhai	zhai	zhai	zhao	zhou	zhan	zhen	zhang	zheng	zhu	zhuo	zhuai			zhui	zhuai	zhun		zhi															
ch	chi	cha		chai	chai	chai	chao	chou	chan	chen	chang	cheng	chu	chuo	chuai			chui	chuan	chun		chi															
sh	shi	sha		shai	shai	shai	shao	shou	shan	shen	shang	sheng	shu	shuo	shuai			shui	shuan	shun		shi															
r	ri			re			rao	rou	ran	ren	rang	reng	ru	ruo			rui	ruan	run		ri																
j	ji																					ji	jia	jie	jiao	jiu	jian	jiang	jin	jing	jiong	ju	jue	juan	jun		
q	qi																					qi	qia	qie	qiao	qiu	qian	qiang	qin	qing	qiong	qu	quan	qun			
x	xi																					xi	xia	xie	xiao	xu	xian	xiang	xin	xing	xiong	xu	xue	xuan	xun		
Medial	a	o	e	ai	ei	ai	ai	ao	ou	an	en	ang	ong	wu	wa	wo	wai	wan	wang	wen	weng	yi	ya	ye	yao	you	yan	yang	yin	ying	yu	yue	yuan	yun			

Gambar 2.1 Chinese pinyin (Quick Mandarin, 2005)

Misalnya, kata 卖 (*Pinyin: Mǎi*) dan 买 (*Pinyin: Mài*). *Mǎi* berarti menjual dan *Mài* berarti membeli. Pengetahuan tentang pelafalan bunyi penting agar pesan yang terkandung di dalam kalimat dapat dimengerti dan tidak menimbulkan kesalahpahaman. Setiap karakter dalam bahasa Cina mempunyai arti tersendiri. Karakter-karakter tersebut juga dapat bergabung dan membentuk suatu kesatuan baru dan mempunyai arti yang berbeda.



Gambar 2.2 Tekanan bunyi dalam pinyin (Hemsida, 2013)

Menurut Swoffod (2010), meskipun karakter Cina direpresentasikan dalam satu silabel, Bahasa Mandarin juga mempunyai suatu kesatuan suku kata. Spasi digunakan setelah suku kata selesai ditulis, bukan per satu silabel. Berikut ini adalah jenis aturan yang dimiliki oleh Bahasa Mandarin.

1. *General*

- a. *Single meaning* yaitu setiap karakter yang merupakan sebuah set yang terdiri sampai dua atau tiga karakter ditulis berdampingan. Contoh: *rén* (人, orang), *péngyou* (朋友, teman), *qiǎokèlì* (巧克力, coklat).
- b. *Combined meaning* (dua karakter) yaitu gabungan karakter-karakter yang membentuk makna yang berbeda. Contoh: *hǎifēng* (海风, angin laut).

Karakter *hǎi* (海, laut) dan *fēng* (风, angin) jika berdiri sendiri mempunyai arti secara berturut-turut yaitu laut dan angin.

- c. *Combined meaning* (empat karakter atau lebih) yaitu gabungan karakter-karakter yang terdiri dari empat karakter atau lebih sehingga mempunyai arti yang berbeda dengan ketika karakter-karakter ini berdiri sendiri. Contoh: *huánjìng bǎohù guīhuà* (环境保护规划, perencanaan perlindungan lingkungan).
2. *Duplicated words*
- a. Pola AA yaitu duplikasi pada karakter yang ditulis bersama-sama. Contoh: *rénrén* (人人, setiap orang), *kànkàn* (看看, melihat-lihat), *niánnián* (年年, setiap tahun).
- b. Pola ABAB yaitu dua karakter yang diduplikasi dan ditulis terpisah. Contoh: *yánjiū* (研究研究, belajar, meneliti), *xuěbái xuěbái* (雪白雪白, putri salju).
- c. Pola AABB yaitu gabungan dua karakter dengan bantuan sebuah tanda penghubung. Contoh: *láilái-wǎngwǎng* (来来往往, pergi), *qiānqiān-wànwàn* (千千万万, banyak).
3. *Nouns and names*. Kata benda yang merupakan gabungan dari beberapa karakter mempunyai aturan untuk ditulis berdampingan. Contoh: *zhuōzi* (桌子, meja). Penggunaan karakter Mandarin untuk nama orang, nama tempat, dan panggilan sapaan mengharuskan huruf pertama dari pinyin ditulis secara kapital. Contoh: *Běijīng Shì* (北京市, kota Beijing). Untuk penggunaan gelar, huruf pertama dari *pinyin* tidak ditulis kapital. Contoh: *Lǐ xiānsheng* (Mr. Li), *Tián zhǔrèn* (direktur Tian).

4. *Verbs*

- a. Kata kerja dan objek ditulis terpisah. Contoh: *kàn xìn* (看信, membaca surat).
- b. Jika kata kerja dan komplementnya terdiri dari satu silabel, maka penulisannya digabung. Contoh: *gǎixiě wéi* (menulis ulang skenario).

5. *Adjectives*

Jika ada sebuah kata sifat ditambah dengan kata duplikasikan ditulis seperti yang dijelaskan pada contoh. Contoh: *liàngtāngtāng* (bersinar terang)

6. *Pronouns*

Kata ganti yang menggunakan imbuhan *-men* ditulis berdampingan. Contoh: *wǒmen* (我们, kami), *tāmen* (他们, mereka).

7. *Numeral and measure words*

- a. Kata yang mempunyai arti setiap, dalam konteks jumlah ditulis terpisah dengan objek yang dihitung. Contoh: *měi nián* (每年, setiap tahun).
- b. Angka sampai dengan 100 ditulis tanpa dipisah. Untuk angka lebih dari 100 ditulis secara terpisah.
- c. Pengurutan angka dilakukan seperti pada contoh. Contoh: *dì-yī* (第一, pertama).

8. *Other function words*

- a. *Adverbs*
- b. *Preposition*
- c. *Conjunction*
- d. *Constructive auxiliaries*
- e. *Modal auxiliaries* pada akhir kalimat

- f. *Exclamations* dan *interjection*
- 9. *Hyphenation*

Penambahan kata penghubung dilakukan pada kondisi di bawah ini.

1. Sebagai penghubung antar objek dan menyatakan perbedaan.
2. Penyingkatan kata
3. Idiom empat suku kata. Frase tradisional Cina yang sebagian besar terdiri dari empat silabel.

Penelitian ini mempunyai batasan yaitu aplikasi kamus hanya akan menggunakan *Simplified Chinese Character* karena penulisan karakter yang lebih sederhana dan karakter jenis ini adalah yang banyak dipelajari saat ini.

2.4 Levenshtein Distance

Algoritma *Levenshtein Distance* atau *Edit Distance* adalah algoritma pencarian jumlah perbedaan *string* yang ditemukan oleh Vladimir Levenshtein, seorang ilmuwan Rusia, pada tahun 1965. Algoritma ini banyak digunakan dalam berbagai bidang, misalnya pencarian *string*, pendeteksi plagiarisme dan *speech recognition*. Algoritma ini merupakan perkembangan dari *dynamic programming* dimana algoritma ini melakukan perbandingan dengan menggunakan matriks pembandingan dan memberi *output* jumlah langkah pembeda di antara dua *string*. Algoritma ini menentukan *distance* berdasarkan jumlah minimum perubahan yang terjadi ketika terjadi transformasi dari bentuk *string* awal ke bentuk *string* lain (Levenshtein, 1965).

Dalam konteks *string matching*, objektivitasnya adalah untuk menyocokkan *string* pendek dengan teks yang lebih panjang. Hal ini akan menyebabkan banyak

timbulnya perbedaan. Perbedaan antar *string* tersebut dihitung dengan algoritma Levenshtein Distance atau yang sering disebut dengan algoritma Edit Distance.

2.4.1 Prinsip Dasar Levenshtein Distance

Algoritma Levenshtein Distance mempunyai tiga prinsip dasar untuk menghitung banyaknya perbedaan antar *string*. Prinsip dasar dari algoritma ini adalah dilakukannya tiga operasi dasar, yaitu penyisipan, penggantian, dan penghapusan. Berikut ini merupakan tiga operasi dasar dari algoritma Levenshtein Distance (Adiwidya, 2009)

1. Penyisipan (*Insertion*)

Insertion adalah operasi melakukan penyisipan sebuah karakter ke dalam sebuah *string* tertentu. Misalnya ada dua buah variabel, yaitu S = “brian” dan T = “barisan”. Proses penyisipan dilakukan dengan menambah huruf a dan s pada posisi 2 dan 5.

	1	2	3	4	5	6	7
T	b	a	r	i	s	a	n
S	b	-	r	-	a	n	
		a		s			

Gambar 2.3 Proses *insertion* (Adiwidya, 2009)

2. Penghapusan (*Deletion*)

Pada proses penghapusan, misalnya S = “memori” dan T = “Meri”. Penghapusan dilakukan untuk karakter m pada lokasi ke-3 dan o pada lokasi ke-4. Dua operasi penghapusan tersebut menunjukkan transformasi S ke T .

	1	2	3	4	5	6
T	m	e	-	-	r	i
S	m	e	m	o	r	i

Gambar 2.4 Proses *deletion* (Adiwidya, 2009)

3. Penggantian (*Substitution*)

Dikatakan bahwa S bernilai “perasa” dan T bernilai “pewara”. Kemudian, *string* ditransformasikan S dengan melakukan substitusi pada posisi ke-3 dan posisi ke-5.

	1	2	3	4	5	6
T	p	e	w	a	r	a
S	p	e	r	a	s	a
			w		r	

Gambar 2.5 Proses *substitution* (Adiwidya, 2009)

2.4.1 Implementasi Algoritma Levenshtein

Di dalam algoritma Levenshtein Distance ada langkah-langkah yang diperhatikan. Algoritma ini dapat menggunakan dua cara, yaitu rekursif dan iterasi. Penelitian ini tidak akan menggunakan rekursif karena proses rekursif hanya akan memberatkan sistem.

Pertama-tama yang dilakukan adalah perhitungan panjang kedua *string*, dengan pemisalan m adalah panjang kata pertama, dan n adalah panjang kata kedua. Penyocokan kedua kata tersebut akan dipecah ke bagian-bagian yang lebih kecil. Fungsi *Levenshtein Distance* menggunakan dua buah *string* dan membandingkannya sehingga menghasilkan suatu nilai *distance* (Adiwidya, 2009). Contoh kasusnya, yaitu kata “kitten” dan “sitting”. Kedua kata ini mempunyai *Levenshtein Distance* sebesar 3.

1. Kitten > Sitten, terjadi substitusi atau penggantian karakter dari “K” menjadi “S”.
2. Sitten > Sittin, terjadi substitusi pada karakter “e” menjadi “i”.
3. Sittin > Sitting, terjadi penyisipan karakter “g” di belakang kata “sittin”.

Pada penerapan Levenshtein Distance, dihasilkan *output* dalam bentuk matriks perkalian sebesar panjang *string* pertama ditambah 1 dikali dengan *string* kedua ditambah 1. Hasil akhir untuk *distance* pada kedua *string* terdapat pada ujung dari matriks yang dihasilkan. Misalnya, untuk *string* pertama, yaitu “kitten” dan *string* kedua, yaitu “sitting”. Tabel 2.1 merupakan contoh matriks perbandingan kata “kitten” dan kata “sitting dengan perbedaan antara kedua *string* sebanyak 3.

Tabel 2.1 Tabel perbandingan "kitten" dan "sitting"

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
I	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

Gambar berikut ini merupakan *pseudocode* dari algoritma Levenshtein Distance yang digunakan pada penelitian ini.

```

int computeDistance(String str1, String str2)
{
    declare m = length of str1
    declare n = length of str2

    declare costs[n+1]

    FOR i from 0 to m
        DECLARE lastValue = i
        FOR j from 0 to n
            IF i = 0 THEN
                costs[j] = j
            ELSE
                IF j IS GREATER THAN 0 THEN
                    DECLARE newValue = costs[j - 1]
                    IF str1.charAt(i - 1) IS NOT EQUAL TO str2.charAt(j - 1) THEN
                        newValue = minimum(newValue, lastValue, costs[j]) + 1
                    END IF
                    costs[j - 1] = lastValue
                    lastValue = newValue
                END IF
            END IF
        END FOR
        IF i IS GREATER THAN 0 THEN
            costs[n] = lastValue
        END IF
    END FOR

    RETURN costs[n]
}

```

Gambar 2.6 *Pseudocode* algoritma Levenshtein Distance

2.5 Android

Android adalah sistem operasi berbasis Linux yang didesain untuk *smartphone* layar sentuh dan *tablet computer*. Android merupakan sistem operasi yang *open-source* yang dikembangkan oleh Android Inc., yang kemudian dibeli oleh Google. Setiap versi pada Android dinamai dengan nama pencuci mulut secara urut menurut alfabet. Versi terakhir sistem operasi Android yang dirilis, yaitu versi 4.3 dengan nama *Jelly Bean*. Pada tanggal 3 September 2013, Android telah mengumumkan akan ada versi 4.4 yang diberi nama *Kitkat*. Android merupakan sistem operasi pada *smartphone* dan *tablet* yang paling banyak terjual menurut sebuah lembaga survey. (Gartner Inc., 2013).



Gambar 2.7 Logo Android (Android, 2013)

Android terdiri dari *kernel* yang berdasar pada Linux kernel 2.6 dan Linux Kernel 3.x (untuk versi Android 4 ke atas). Android dilengkapi dengan *middleware*, *libraries*, dan API yang ditulis dalam bahasa C dan *application software* yang berjalan dalam suatu *application framework*, termasuk *library-library* yang kompatibel dengan Java. Android mengeksekusi programnya di atas Dalvik *virtual machine*, yang melakukan kompilasi terhadap Java *byte code*.

2.6 SQLite

SQLite adalah sebuah sistem manajemen basis data relasional yang ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat *public domain* yang dikembangkan oleh D. Richard Hipp.

SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada SQL-92, termasuk transaksi bersifat *atomic*, konsistensi basis data, isolasi, dan durabilitas, *trigger*, dan *query-query* yang kompleks. Tidak ada pengecekan tipe, sehingga data bisa dimasukkan dalam bentuk *string* untuk sebuah kolom bertipe data *integer*. Beberapa proses atau *thread* dapat berjalan secara bersamaan dan mengakses basis data yang sama tanpa mengalami masalah. Hal ini disebabkan oleh akses pembacaan data yang dilakukan secara paralel. Sementara itu, untuk penulisan ke dalam basis data hanya bisa dilakukan jika tidak ada yang sedang menulis ke dalam basis data.



Gambar 2.8 SQLite (SQLite, 2013)

Perancangan basis data untuk *platform* Android dapat dilakukan melalui PC dengan menggunakan peranti lunak tambahan. Adapun peranti lunak yang disediakan ada yang cuma-cuma dan ada juga yang mengharuskan pengguna untuk membayar agar dapat menggunakan fitur-fitur peranti lunak secara lengkap. Beberapa peranti lunak yang dapat digunakan untuk membuat basis data SQLite, diantaranya SQLite Manager, SpatiaLite, SQLite Database Browser, SQLiteMan,

SQLiteStudio, SQLite Administrator, SQLite Expert, dan SQLite Professional. Penelitian ini menggunakan basis data SQLite yang dirancang dengan peranti lunak SQLite Database Browser .

