

Metode Pencocokan String dengan Transliterasi Nama dalam Aksara Arab dan Latin

Abdurrahman Dihya Ramadhan / 13509060

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

a.dihya@students.itb.ac.id

Abstraksi—Pattern Matching, atau pencocokan pola merupakan algoritma yang digunakan untuk pencarian maupun validasi suatu teks. Pada zaman ini telah berkembang banyak algoritma *pattern matching*. Pada zaman ini penggunaan mesin pencari telah berkembang pesat. Kebutuhan untuk melakukan pencarian kata atau istilah dalam teks, baik di dalam komputer maupun internet meningkat. Seiring bertambah luasnya pemakaian internet, maka pencarian dengan menggunakan berbagai bahasa pun mutlak harus dilakukan. Saat ini pun telah berkembang algoritma untuk mencari kata dengan memasukkan kata yang berbeda bahasa dengan onjek pencariannya. Ini dinamakan transliterasi. Salah satu transliterasi yang sering digunakan dan dijadikan bahan penelitian adalah transliterasi arab dan latin. Bahasa Arab mempunyai tulisan yang unik karena setiap huruf terdiri dari huruf dan tanda baca. Dengan menggunakan algoritma pencocokan string, penerapan kaidah transliterasi arab ke latin yang telah disahkan, dan metode lain dalam pencarian, kita dapat melakukan pencarian transliterasi arab ke latin dan sebaliknya dengan efektif.

Term Indeks—Fuzzy string matching, Soundex, Levenshtein edited distance, Editex, SecondString, Baseline Task, Huruf Hijaiyah.

I. PENDAHULUAN

Metode pencocokan nama adalah metode yang berkembang seiring dengan kebutuhan transliterasi antar bahasa. Nama yang dimaksud bisa berupa nama orang, nama daerah, nama sebuah komunitas, dan sebagainya. Pencocokan nama yang masuk ke dalam pembahasan ini adalah pencocokan nama yang tidak terikat dengan bahasa. Seperti kita ketahui bahwa nama seseorang di berbagai bahasa tidaklah berubah penyebutannya. Namun secara tulisan berubah sesuai dengan tulisan yang digunakan.

Pencocokan nama menjadi masalah yang dibahas dalam pemrosesan konten – konten linguistik oleh teknologi pemrosesan bahasa. Contohnya adalah penarikan informasi, pengelompokan dokumen, mesin pencarian, dan translasi.

Di antara bahasa-bahasa di dunia, bahasa arab memiliki keunikan tersendiri dalam penulisan bahasanya. Berikut ulasan singkat mengenai penulisan bahasa arab.

1.1 Huruf Hijaiyah

Huruf hijaiyah adalah huruf-huruf dalam bahasa arab. Huruf-huruf ini telah digunakan ribuan tahun yang lalu. Sampai saat ini, berbagai bahasa menggunakan huruf hijaiyah untuk penulisannya, seperti bahasa Farsi, Urdu, dan sebagainya.

Huruf hijaiyah ini juga merepresentasikan huruf konsonan. Dalam transliterasi huruf arab ke huruf latin, huruf hijaiyah direpresentasikan sebagai huruf mati. Maka kalimat فتح yang seharusnya dibaca *fataha* menjadi *fth* setelah proses transliterasi. Hal demikian karena dalam tulisan tersebut tidak terdapat tanda baca yang menunjukkan huruf vokal. Tabel berikut menunjukkan jenis – jenis huruf hijaiyah dalam bahasa arab.

Huruf	Istilah	Huruf	Istilah
ا	Alif	ط	Tha'
ب	Ba'	ظ	Zha'
ت	Ta'	ع	'ain
ث	Tsa'	غ	Ghain
ج	Jim	ف	Fa'
ح	Ha' (tipis)	ق	Qaf
خ	Kha'	ك	Kaf
د	Dal	ل	Lam
ذ	Dzal	م	Mim
ر	Ra'	ن	Nun
ز	Zay	و	Waw
س	Sin	ه	Ha' (tebal)
ش	Syin	ء	Hamzah
ص	Shad	ي	Ya'
ض	Dhad		

Tabel 1: Huruf hijaiyah dan istilah sebutannya

1.2 Tanda baca

Tanda baca disebut juga *syakl* dalam bahasa arab. Atau sering disebut juga harakat. Tanda baca inilah yang merepresentasikan huruf vokal dalam bahasa arab. Tulisan arab bias menggunakan harakat bias pula tidak. Untuk mengetahui cara membaca tulisan yang tidak menggunakan harakat kita perlu mengetahui tata bahasa

dalam bahasa arab terlebih dahulu. Contohnya kata بيت yang berarti rumah. Kata tersebut bias sacara dibaca baytun, biyta, ataupun baytin. Namun dengan mengetahui makna kata tersebut dan kedudukannya dalam kalimat, kita dapat mengetahui bahwa cara pembacaanya yang benar adalah baytun. Jika ditulis dengan tanda baca, maka penulisan kata tersebut akan menjadi بَيْتٌ. Jelaslah bahwa bahasa arab memiliki aturan sendiri yang unik dalam masalah penulisan bahasa.

Dalam proses translasi nama dari satu tulisan ke tulisan yang lain dimungkinkan adanya perbedaan kombinasi representasi dari huruf mati maupun huruf vokal dari bahasa lain. Penerjemahan dari tulisan arab menjadi tulisan latin berbahasa Indonesia mungkin sudah distandarisasi dengan pedoman transliterasi arab – latin yang disepakati Departemen Agama. Namun dalam mesin pencarian tetap saja ada kata yang semakna namun memiliki ejaan yang berbeda karena kata tersebut telah menjadi kata serapan. Contohnya حكم jika ditulis dengan huruf latin menjadi *hukm* namun kata serapannya adalah hukum. Atau *العصر* yaitu al-‘ashru yang kata serapannya asar. Dan banyak lagi contoh lainnya.

Variasi ini menyebabkan tulisan bahasa arab dapat direpresentasikan menjadi beberapa tulisan latin. Apabila kemungkinan – kemungkinan serapan kata dimasukkan juga ke dalam suatu *library* atau penampung data, maka dikhawatirkan terjadi kebocoran memori.

Dengan algoritma pencocokan string kita dapat mencari kata yang cocok dan sepadan dari bahasa A ke bahasa B termasuk mendaftar kemungkinan – kemungkinan serapan

kata tersebut dalam bahasa lain. Beberapa algoritma pencocokan yang telah umum biasanya mensyaratkan suatu *pattern* atau pola string yang direpresentasikan oleh kumpulan karakter yang umum digunakan. Namun pada makalah ini solusi yang ditawarkan adalah dengan prosedur normalisasi dan pemetaan kesamaan untuk pengukuran jarak standar Levenshtein.

II. PEDOMAN TRANSLITERASI ARAB-LATIN INDONESIA

Jika kita melakukan pencarian sumber dari bahasa arab ke bahasa Indonesia melalui internet, maka banyak digunakan kata yang berasal dari bahasa Arab dengan aneka ragam lafal dan tulisan walaupun berasal dari kata yang sama. Maka disusunlah suatu pedoman untuk menunjukkan perbedaan itu agar perbedaan tersebut dapat dipahami. Walaupun banyak variasi dalam penulisan kata dari bahasa Arab, hendaknya kata yang populer diutamakan penggunaannya.

Kata dalam bahasa arab selain dapat ditransliterasikan ke dalam tulisan latin tetapi banyak juga kata dalam bahasa arab yang sudah menjadi kata serapan dalam bahasa Indonesia. Berikut adalah aturan transliterasi bahasa arab ke dalam aksara latin beserta contoh serapannya.

No.	Penulis an Arab	Alih aksara kritis	Alih aksara diplom atik[4]	Peruba han	Kata dari alih aksara kritis	Kata serapan
1.a.	ع	'a	'a	A	Assalamu'alaykum, 'alayhissalam, syari'at, 'Ashr, 'Abdullah, 'Abdul Muththalib, 'Aisyah, 'Amr	Assalamualaikum, alaihissalam, syariat, Ashar, Abdullah, Abdul Muttalib, Aisyah, Amr
b.	ع	'i	'i	I	'Isa, 'Isya, 'Idul Fithri, 'Idul Adhha, al-'Iraq, dhu'afa', dha'if, adh-Dha'ifah	Isa, Isya, Idul Fitri, Idul Adha, Irak, duafa, dhaif, adh-Dhaifah
c.	ع	'u	'u	U	'Umar ibn al-Khaththab, 'Utsman ibn 'Affan, 'ulama`	Umar bin Khattab, Utsman bin Affan, ulama
2.	ء	` atau '	tidak dilamb ang-		Al-Qur'an, an-Nasa'iy	Al-Quran, an-Nasai
3.	ئ	` atau '	tidak dilamb ang-		Isra', 'Isya`, 'ulama`, dhu'afa`, Muwaththa'	Isra, Isya, ulama, duafa, Muwatta
4.	ي	iy	'a	I	Yahudiyy, Nashraniyy, nabiyy, kursiyy, al-Khudriyy, al-Bukhariyy, an-Nasa'iy	Yahudi, Nasrani, nabi, kursi, al-Khudri, al-Bukhari, an-Nasai
5.a.	اَلْ	al-	al-	diabaik an	Al-Qur'an, Al-'Iraq, an-Nawawiyy, al-Albaniyy, al-Qardhawiyy	Quran, Irak, Nawawi, Albani, Qardhawi
b.	اَلْ	al-	al-	Al- (ditulis bersam	Al-Kitab, Al-Qur'an	Alkitab, Alquran

				bung)		
6.a.	ص	sh	sh	S	Masjidil Aqsha, Bashrah, ikhlash, shadaqah, shahih, shalat, shubh, 'ashr	Masjidil Aqsha, shahih, shalat, shubuh, ashar
b.	ث	ts	ts	Ts	hadits, 'Utsman	hadis, Usman
7.	ذ	dz	dz	Dz	adzab, adzan, muadzin, madzhab, at-Tirmidzi	adzab, adzan, muadzin, madzhab, Tirmidzi
8.	ظ	zh	zh	Z	zhahir, zhalim, zhuhur	zhahir, zhalim, zhuhur
9.	ظ	zh	zh	Zh	hafazh, nazhar, zhahir, zhalim, zhuhur	hafal, nalar, lahir, lalim, lohor
10.	ض	dl	dl	DI	dlu'afa`, haidl, ridla, Ramadhan, 'Idul Adlha	dluafa, haidl, ridla, Ramadhan, Idul Adlha,
11.	ط	th	th	T	'Abdul Muththalib, 'Umar ibn al-Khaththab, Fathimah	Abdul Muttalib, Umar bin Khattab, Fatimah
12.	ف	f	f	P	fahm, nafs	paham, napas
13.	ق	q	q	K	Ya'qub, al-'Iraq, aqidah, akhlaq, fiqh, haqiqah	Ya'kub, Irak, akidah, akhlak, fikih, hakikat
14.	غ	'	'	K	Ja'far, jama', da'wah, Mi'raj, ma'ruf, ma'shiyyat, mu'jizat, ta'dil	Jakfar, jamak, dakwah, Mikraj, makruf, maksiat, mukjizat, takdil
15.	ؤ	'	'	K	mu'min, ru'yat	mukmin, rukyat
16.a.	ع	aw	au	Au	Fir'awn, Sawdah	Firaun, Saudah
b.	ي	ay	ai	Ai	Al-Layl, Layla, Assalamu'alaykum, 'alayhissalam, bayt	Al-Lail, Laila, Assalamualaikum, alaihissalam, bait
17.a.	هـ	a	a	E	Husayn, jama'ah, Makkah, Madinah, masjid, shadaqah, syaikh	Husein, jemaah, Mekkah, Medinah, mesjid, sedekah, syeikh

Tabel 2: Tabel perbandingan alih aksara dan kata serapan

III. FUZZY STRING MATCHING

Algoritma yang dipakai untuk menyelesaikan masalah transliterasi dan pencarian bahasa Arab ke bahasa Indonesia adalah dengan *fuzzy string matching*. Istilah pencocokan string dengan metode fuzzy biasanya digunakan untuk menggambarkan suatu metode yang mencocokkan string dengan lebih berdasarkan kepada kesamaan pola ketimbang identitas dari string itu sendiri. Metode fuzzy yang umum digunakan meliputi pengaturan jarak, pencocokan n-gram, dan prosedur normalisasi seperti soundex.

A. Soundex

Berdasarkan namanya, prosedur Soundex ini merupakan prosedur pencocokan string berdasarkan suara dari pelafalan kata. Prosedur ini dipatenkan oleh Odell dan Russel dan telah dirancang untuk mencari variasi ejaan dari suatu nama. Soundex merepresentasikan kelas - kelas dari suara yang dapat diucapkan bersama – sama dari suatu tulisan pada bahasa tertentu. Algoritma dan kelas yang digunakan pada prosedur ini ditunjukkan pada gambar 1 dan 2.

Code: 0 1 2 3 4 5 6
Letters: aeiouy bp cgjqk dt l mn r
hw fv sxz

Gambar 1: Kode fonetik pada soundex

1. Replace all but the first letter of the string by its phonetic code.
2. Eliminate any adjacent repetitions of codes.
3. Eliminate all occurrences of code 0, i.e. eliminate all vowels.
4. Return the first four characters of the resulting string.
5. Examples: Patrick = P362, Peter = P36, Peterson = P3625

Gambar 2: Algoritma soundex

Contoh dalam gambar 2 menunjukkan bahwa beberapa nama yang berbeda dapat muncul dan dianggap cocok satu sama lain menggunakan algoritma soundex.

B. Levenshtein Edit Distance

Algoritma ini adalah algoritma untuk mencari nilai jarak perubahan suatu string. Algoritma Levenshtein mengukur jarak perubahan di mana jarak perubahan didefinisikan sebagai jumlah pemasukka, penghapusan, atau penggantian huruf yang diperlukan untuk membuat dua string menjadi cocok. Nilai nol menunjukkan bahwa

kedua string identik.

Dimisalkan ada dua string, string s dengan ukuran m dan string t dengan ukuran n . maka algoritma ini mempunyai $O(nm)$ kompleksitas waktu dan tempat. Dalam algoritma ini dibuat matriks dengan n baris dan m kolom. Terdapat pula fungsi $e(s_i, t_j)$ di mana s_i adalah karakter dalam string s , dan t_j adalah karakter dalam string t . Fungsi ini mengembalikan nilai 0 apabila kedua karakter sama. Algoritma ini dapat direpresentasikan dengan hubungan rekurens yang tampak pada gambar berikut.

```

for each i from 0 to |s|
  for each j from 0 to |t|
    levenshtein(0; 0) = 0
    levenshtein(i; 0) = i
    levenshtein(0; j) = j
    levenshtein(i; j) =
      min[levenshtein(i - 1; j) + 1;
          levenshtein(i; j - 1) + 1;
          levenshtein(i - 1; j - 1) +
            e(si; tj)]

```

Gambar 3: Hubungan rekurens pada jarak perubahan Levenshtein

Algoritma pencocokan fuzzy yang sederhana dapat dibuat dengan memisahkan nilai jarak perubahan Levenshtein dengan panjang dari string yang terpanjang atau terpendek, mengurangi nilai ini dengan satu, kemudian mengatur nilai awal yang harus didapat untuk menyatakan sebuah string cocok. Dengan pendekatan sederhana ini, pasangan string yang lebih panjang lebih memungkinkan untuk dianggap cocok dari pada pasangan string yang lebih pendek dengan jumlah karakter yang berbeda yang sama.

C. Editex

Algoritma editex dijelaskan pertama kali oleh Zobel dan Dart (1996). Algoritma ini mengombinasikan antara soundex dan levenshtein dengan mengganti fungsi $e(s_i, t_i)$ dengan fungsi $r(s_i, t_i)$. Fungsi $r(s_i, t_i)$ mengembalikan nilai 0 apabila dua huruf berada pada kelompok yang sama. Selain itu fungsi tersebut mengembalikan 2. Algoritma selengkapnya ditunjukkan pada gambar 4 dan 5. Algoritma editex menetralkan h dan w . Dalam algoritma pada gambar, hal ini ditunjukkan oleh $d(s_i, t_i)$. Fungsi ini sama dengan $r(s_i, t_i)$ dengan dua pengecualian yang disebutkan di atas. Fungsi hanya tersebut membandingkan huruf dari string yang sama dan tidak perlu membandingkan string yang berbeda. Perbedaan yang lain adalah apabila s_{i-1} adalah h atau w , dan $s_{i-1} \neq s_i$, maka $d(s_i, t_i)$ bernilai 1.

```

for each i from 0 to |s|
  for each j from 0 to |t|
    editex(0; 0) = 0
    editex(i; 0) = editex(i - 1; 0) + d(si-1; si)
    editex(0; j) = editex(0; j - 1) + d(tj-1; tj)
    editex(i; j) = min[editex(i - 1; j) +
                      d(si-1; si);
                     editex(i; j - 1) + d(tj-1; tj);
                     editex(i - 1; j - 1) + r(si; tj)]

```

Gambar 4: Hubungan rekurens dari jarak perubahan Editex

0 1 2 3 4 5 6 7 8 9

aeiouy bp ckq dt lr mn gj fpv sxz csz

Gambar 5: Pengelompokkan huruf Editex

Zobel dan Dart, pada tahun 1996 mendiskusikan beberapa perbaikan pada algoritma pencocokan string dengan Soundex maupun Levenshtein. Salah satu perbaikan yang dilakukan adalah dengan apa yang mereka sebut “peruncingan”. Peruncingan ini mencakup pemberian bobot yang lebih tinggi apabila ketidakcocokan terjadi di awal string. Dan apabila ketidakcocokan terjadi di string string yang lebih akhir, maka bobot yang diberikan lebih rendah. Perbaikan lainnya adalah sesuatu yang mereka sebut dengan “metode fonometrik”, yaitu ketika setiap string masukan dipetakan berdasarkan representasi fonemik yang berdasarkan pelafalan. Kemudian algoritma jarak perubahan diterapkan pada representasi fonemik dari string.

Penamaan dalam bahasa Arab untuk nama selain nama Arab tidak selalu dapat diprediksi. Berikut aturan-aturan untuk merepresentasikan nama asing dalam bahasa Arab.

1. Hamzah atau ‘ain akan sering muncul dalam teks latin sebagai apostrof atau huruf vokal yang mengikutinya.
2. Nama selain nama arab akan memiliki huruf vokal yang terdiri dari alif, ya, dan waw, untuk merepresentasikan huruf vokal.
3. Tulisan romawi “p” dan “v” akan direpresentasikan dengan “b” dan “f” dalam bahasa arab.
4. Kebanyakan nama arab tidak memiliki huruf vokal dalam representasinya.
5. “c” direpresentasikan dengan “j” atau “sy”.

D. Normalisasi. String Berbahasa Arab

Diperlukan adanya pemetaan masing-masing huruf dalam bahasa arab ke dalam huruf transliterasi. Karakter-karakter tersebut ditransformasikan ke dalam representasi ekspresi regular sebagai berikut.

Transliterasi	Ekuivalensi dalam Kelas Aksara Latin	Huruf Arab
‘	‘,a ,A,e,E,i,I,o,O,u,U	ء
	‘,a ,A,e ,E,i ,I,o ,O,u,U	ا
>	‘,a ,A,e ,E,i ,I,o ,O,u,U	ا
&	‘,a ,A,e ,E,i ,I,o ,O,u,U	و
<	‘,a ,A,e ,E,i ,I,o ,O,u,U	!

}	'a ,A,e ,E,i ,I,o ,O,u ,U	ئ
A	'a ,A,e ,E,i ,I,o ,O,u ,U	ا
b	b ,B,p ,P,v,V	ب
p	a ,e	ة
+	a ,e	ة
t	t,T	ت
v	t ,T	ث
j	j,J,g,G	ج
H	h, H	ح
x	k, K	خ
d	d, D	د
*	d, D	ذ
r	r, R	ر
z	z, Z	ز
s	s, S,c, C	س
\$	s, S	ش
S	s, S	ص
D	d, D	ض
T	t, T	ط
Z	z, Z,d, D	ظ
E	' ,c,a,A,e,E,i,I,o,O,u,U	ع
`	' ,c,a,A,e,E,i,I,o,O,u,U	ع
g	g, G	غ
f	f, F,v, V	ف
q	q, Q, g, G,k, K	ق
K	k, K,c, C,S, s	ك
L	l, L	ل
M	m, M	م
N	n, N	ن
H	h, H	ه
W	w, W,u, u,o, O, 0	و
Y	y, Y, i, I, e, E, ,j, J	ي
Y	a, A,e, E,i, I, o,O,u, U	ى
A	a, e	َ

Tabel 3: Ekuivalensi Aksara Arab ke latin

```

$S1 =~ s/$/sh/g; # normalize Buckwalter
$S1 =~ s/v/th/g; # normalize Buckwalter
$S1 =~ s/*/dh/g; # normalize Buckwalter
$S1 =~ s/x/kh/g; # normalize Buckwalter
$S1 =~ s/(F|K|N|o|~)//g; # remove case vowels,
# the shadda and the sukuun
$S1 =~ s/'aa^/g; # normalize basis w/
# Buckwalter madda
$S1 =~ s/(U|W|I|A)/A/g; # normalize hamza
$S1 =~ s/_//; # eliminate underscores
$S1 =~ s/s//g; # eliminate white space

```

Gambar 5: Normalisasi String Arab

Perkembangan metode normalisasi dimaksudkan agar string dalam bahasa latin semakin mirip dengan transliterasi bahasa arabnya. Implementasi dari normalisasi bahasa latin adalah seperti ditunjukkan oleh gambar berikut.

```

$S2 =~ s/(a|e|i|A|E|I)(e|i|y)/y/g;
# hi diphthongs go to y in Arabic
$S2 =~ s/(e|a|o)(u|w|o)/w/g;
# lo diphthongs go to w in Arabic
$S2 =~ s/(P|p)h/f/g; # ph -> f in Arabic
$S2 =~ s/(S|s)ch/sh/g; # sch is sh
$S2 =~ s/(C|c)h/tsh/g; # ch is tsh or k ,
# we catch the "k" on the pass
$S2 =~ s/-//g; # eliminate all hyphens
$S2 =~ s/x/ks/g; # x->ks in Arabic
$S2 =~ s/e( |$)/$1/g; # the silent final e
$S2 =~ s/(\S)\1/$1/g; # eliminate duplicates
$S2 =~ s/(\S)gh/$1/g; # eliminate silent gh
$S2 =~ s/s//g; # eliminate white space
$S2 =~ s/(\.|\,|;)//g; # eliminate punctuation

```

Gambar 6: Normalisasi string bahasa latin

Berikut contoh representasi nama latin ke dalam bahasa arab.

Nama Latin	Nama Arab	Transliterasi Arab
Bill Clinton	بيل كلينتون	byl klyntwn
Collin Powel	كولين باول	kwlyn bAwl
Richard Cheney	ريشارد تشيني	rytshAr tshyny

Tabel 4: nama latin direpresentasikan dengan tulisan arab

Adapun contoh nama arab yang direpresentasikan dengan aksara latin adalah sebagai berikut.

Nama Arab	Transliterasi Arab	Nama Latin
مصطفى الشيخ ديب	mSTfY Alshykh dyb	Musthafa asy-Syeikh Dib
محمد عاطف	mHmd EATf	Muhammad Athif
حسني مبارك	Hsny mbArk	Husni Mubarak

Tabel 5: nama arab dalam representasi tulisan latin

V. KESIMPULAN

Dengan metode ini pencarian lintas bahasa dan lintas aksara menjadi lebih mudah dan lebih cepat. Namun representasi arab ke latin tidak selalu sama dalam berbagai bahasa. Representasi bahasa inggris dalam hal ini tentu berbeda dengan bahasa Indonesia. Sehingga setiap bahasa memerlukan fungsi sendiri untuk transliterasinya. Meskipun kendala-kendala berupa kesesuaian bahasa dan adanya kasus-kasus unik yang tidak sesuai menjadikan perlunya algoritma ini untuk dikembangkan lebih lanjut.

REFERENSI

- [1] Wikipedia:Pedoman alih aksara Arab ke Latin, (Online), (http://id.wikipedia.org/wiki/Wikipedia%3APedoman_alih_aksara_Arab_ke_Latin, diakses 8 Desember 2011).
- [2] K, Bar, Choueka, Y, N, Dershowitz, *An Arabic to English Example-Cased Translation System*. <http://www.mt-archive.info/ICTIS-2007-Bar.pdf> (Online). Tanggal akses 8 Desember 2011.

- [3] Munir, Rinaldi. Diktat Kuliah IF3051 Strategi Algoritma. 2009. Teknik Informatika ITB : Bandung.
- [4] Buckwalter, Tim. 2002. Arabic Transliteration. <http://www.qamus.org/transliteration.htm>. Tanggal akses 9 Desember 2011.
- [5] Wightwick, Jane.Gaagar, Mahmoud. 2009. Mastering Arabic. New York: Hippocrene Books, Inc.

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, 9 Desember 2011

ttd



Abdurrahman Dihya R.
13509060