

APLIKASI KAMUS BAHASA INDONESIA-ARAB BERBASIS ANDROID MENGGUNAKAN METODE BOYER-MOORE

Eva Darnila¹, Safwandi,² Nanda Fadisar,³
Program Studi Teknik Informatika
Universitas Malikussaleh
Safwandihasanunimal100@gmail.com

ABSTRAK

Kamus merupakan alat bantu yang digunakan untuk menerangkan makna kata-kata yang berfungsi membantu seseorang dalam menemukan makna kata-kata yang baru atau kata-kata yang belum diketahui makna kata tersebut. Bahasa Arab merupakan salah satu bahasa yang digunakan oleh seluruh umat muslim di dunia untuk membaca Al-Qur'an dan berkomunikasi. Penggunaan kamus bahasa Arab – Indonesia sangat sulit karena harus membawa kamus yang berbentuk buku yang memiliki ketebalan dan cukup berat, untuk itu diperlukan Sebuah Aplikasi untuk membantu menterjemahkan bahasa arab ke dalam bahasa indonesia. Aplikasi tersebut berupa kamus mobile, yang dapat dipasang pada perangkat mobile berbasis android. Android cenderung digunakan dalam tablet pc dan smartphone pada zaman sekarang ini, selain itu android mudah dikembangkan karena bersifat open source. Dan untuk mempermudah penterjemahan kata Bahasa arab ke bahasa indonesia, aplikasi ini dirancang dengan menggunakan algoritma boyer-moore. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian string yang ditemukan sebelumnya, Algoritma Boyer-Moore bekerja dengan memulai pencocokan pattern dari kanan bukan dari kiri. Dengan memulai pencocokan karakter dari kanan, maka akan lebih banyak informasi yang didapat. Hasil dari penelitian ini didapatkan dari proses pencocokan kata yang diinput user dengan kata didalam database dan menampilkan hasil penterjemahan kata.

Kunci : Kamus, Android, Bahasa Arab-Indonesia, boyer-moore

1. PENDAHULUAN

Bahasa merupakan Suatu isyarat untuk berkomunikasi. Di dunia terdapat banyak bahasa menjadi ciri khas Negara tersebut, misalnya bahasa Arab yang merupakan salah satu bahasa yang digunakan oleh seluruh umat muslim di dunia untuk membaca Al-Qur'an dan berkomunikasi.

Komunikasi antar manusia memang harus dilakukan karena manusia membutuhkan interaksi dengan manusia lain. Proses komunikasi ini pada kondisi tertentu tidak bisa dilakukan karena kendala bahasa. Banyak informasi yang disampaikan orang lain dalam bahasa yang berbeda, baik dalam bentuk lisan maupun tulisan. Sehingga sangat diperlukan sebuah kamus yang dapat menerjemahkan dari sebuah bahasa ke bahasa yang lain.

Kamus bisa berupa buku dan ukurannya pun bermacam-macam. Ada yang kosa katanya sangat banyak, tetapi untuk membawanya kemana-mana menjadi sangat susah karena ukurannya yang terlalu besar, berat dan tebal. (Rusdi, 2014).

Android adalah sistem operasi berbasis Linux yang dapat digunakan di berbagai perangkat mobile. Android memiliki tujuan utama untuk memajukan inovasi peranti telepon bergerak agar pengguna mampu mengeksplorasi kemampuan dan menambah pengalaman lebih dibandingkan dengan platform mobile lainnya. Hingga saat ini Android terus berkembang, baik secara sistem maupun aplikasinya.

Dengan berkembangnya teknologi smartphone saat ini, khususnya Android semakin mempermudah pengguna untuk mengakses dan mendapatkan informasi. Oleh karena itu pada tugas akhir ini penulis tertarik untuk membuat sebuah "*Aplikasi kamus bahasa Indonesia - Arab Berbasis Android menggunakan metode Boyer-moore*". yang dapat membantu mempermudah dalam menerjemahkan kata bahasa Indonesia kedalam bahasa arab.

2. Algoritma Boyer-Moore

Metode Boyer-Moore adalah algoritma pencarian string yang dipublikasikan pertama kali oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian *string* yang ditemukan sebelumnya, Algoritma Boyer-Moore bekerja dengan memulai pencocokan *pattern* dari kanan bukan dari kiri. Dengan memulai pencocokan karakter dari kanan, maka akan lebih banyak informasi yang didapat.

2.1 Cara kerja

Secara sistematis, langkah-langkah yang dilakukan algoritma Boyer-Moore pada saat mencocokkan *string* adalah:

1. Algoritma Boyer-Moore mulai mencocokkan *pattern* pada awal teks.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - a. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*).
 - b. Semua karakter di *pattern* cocok kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* dengan memaksimalkan nilai penggeseran *good suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks. (Riyanarto Sarno, 2012, 11) .

```
Procedure preBmBc (  
  input p : array [0..n-1] of char,  
  input n : integer,  
  input/output bmBc : array[0..ASIZE-1] of integer  
)
```

Declarasi:

 i= integer

Algoritma:

 for(i:=0 to ALPHABETSIZE-1

```

    bmBc[i]:=n
  endfor
  for(i:=0ton-2)
    bmBc[P[i]]:=n-i-1
  endfor

```

Pseudocode Penghitungan Tabel bmBc

```

Procedure BM(
input m,n : integer,
input P : array [0..n-1] of char,
input T : array [0..m-1] of char,
ouput ketemu : array[0..m-1] of boolean
)

```

Declarasi:

```

  I, j, shift, bmBcShift, bmGsShift : integer
  BmBc : array[0.. ALPHABETSIZE] of integer
  BmGs : array[0..n-1] of integer

```

Algoritma:

```

  preBmBc : (n, P, BmBc)
  preBmGs : (n, P, BmGs)
  i:=0
  while (i<= m-n)do
    j:=n-1
    while (j>=0 n and T(i+j) = P[j])do
      j:=j-1
    endwhile
    if(j<0)then
      ketemu[i]:=true
      shift := bmGs[0]
    else
      bmBcShift := BmBc[chartoint(T[i+j])]-n+j+1
      bmGsShift := BmGs[j]
      shift := max(bmBcShift, bmGsShift)
    endif
    i:= i+shift
  endwhile

```

Pseudocode Algoritma Boyer-Moore

2.2 Pencarian dengan Algoritma *Boyer-Moore*

1. Buat tabel pergeseran *pattern* yang dicari (P) dengan pendekatan *Match Heuristic* (MH) dan *Occurence Heuristic* (OH), untuk menentukan jumlah pergeseran yang akan dilakukan jika mendapat karakter tidak cocok pada proses pencocokan dengan teks (T).
2. Jika dalam proses perbandingan terjadi ketidakcocokan antara pasangan karakter pada *pattern* dan karakter teks, pergeseran dilakukan dengan memilih salah satu nilai pergeseran dari dua tabel, dan memiliki nilai pergeseran paling besar dari tabel *Match Heuristic* dan *Occurence Heuristic*.
3. Dua kemungkinan penyelesaian dalam melakukan pergeseran *pattern*, Jika karakter yang tidak cocok, tidak ada pada *pattern* maka pergeseran adalah sebanyak jumlah karakter pada *pattern*. dan jika karakter yang tidak cocok, ada pada *pattern*, maka banyaknya pergeseran bergantung dari nilai pada tabel *Match Heuristic* dan *Occurence Heuristic*.
4. Jika karakter pada teks yang sedang dibandingkan cocok dengan karakter pada *pattern*, maka posisi karakter pada *pattern* dan teks diturunkan sebanyak 1 posisi, kemudian lanjutkan dengan pencocokan pada posisi tersebut dan seterusnya. Jika kemudian terjadi ketidakcocokan karakter *pattern* dan teks, maka pilih nilai pergeseran terbesar dari tabel *match heuristic* dan nilai tabel *occurence heuristic* (Ramadhansyah,2013).

Cara menghitung tabel *occurrence heuristic*

Contoh *pattern* : NANDA

Panjang *pattern* : 5 karakter

Tabel 2.1 *Occurrence Heuristic*

Indeks	0	1	2	3	4
Pattern	N	A	N	D	A
OH					

Langkah-langkah pemberian nilainya adalah sebagai berikut :

1. Lakukan perhitungan, $OH = (\text{length} - 1 - \text{index})$ length = panjang karakter =5
2. Karakter pertama adalah "N" dengan Index = 0 $OH = (5 - 1 - 0 = 4)$ maka nilai karakter "N" = 4
3. Karakter kedua adalah "A" dengan index = 1 $OH = (5 - 1 - 1 = 3)$ maka nilai karakter "A" = 3
4. Karakter ketiga adalah "N" dengan index = 2 $OH = (5 - 1 - 2 = 2)$ maka nilai karakter "N" = 2
5. Karakter keempat adalah "D" dengan index = 3 $OH = (5 - 1 - 3 = 1)$ maka nilai karakter "D" = 1
6. Karakter kelima adalah "A" dengan index = 4 $OH = (5 - 1 - 4 = 0)$ maka nilai karakter "A" = 0
7. $OH = (5 - 1 - 4 = 0)$ maka nilai karakter "A" = 0

Tabel 2.2 Hasil pencarian Occurrence Heuristic

Indeks	0	1	2	3	4
Pattern	N	A	N	D	A
OH	4	3	2	1	0

Cara menghitung tabel *match heuristic*

Contoh pattern : NANDA

Panjang pattern : 5 karakter

Tabel 2.3 match heuristic

Indeks	0	1	2	3	4
Pattern	N	A	N	D	A
MH					

Langkah-langkah pemberian nilainya adalah sebagai berikut :

1. Berikan nilai 1 (nilai default MH) pada index terbesar, dalam kasus ini index terbesar adalah "A".

2. Bandingkan karakter kedua, jika karakter "A" sudah ditemukan/sudah cocok, tetapi karakter sebelum "A" bukan "D", maka check apakah karakter yang sudah cocok pada teks ada pada *pattern* berikutnya, Jika ada sejajarkan karakter tersebut dengan karakter yang sudah cocok, dan beri nilai pergeseranya ke tabel MH, pada kasus ini "A" sudah cocok.

Tabel 2.4 Proses pergeseran *match heuristic*

Teks	N	A	N	D	A
Pattern	N	A	N	L	A

Tabel 2.5 Proses pergeseran *match heuristic*

Teks	N	A	N	D	A		
Pattern	1	2	N	A	N	L	A

3. Lanjutkan langkah ke dua untuk karakter berikutnya sampai akhir *pattern*, ketidakcocokan *pattern* maka bandingkan karakter yang sudah cocok dengan karakter yang belum dibandingkan pada *pattern*, jika ada maka sejajarkan karakter tersebut, jika tidak maka pergeseranya sejauh panjang karakter, dan beri nilai pergeserannya ke tabel MH.

Tabel 2.6 Hasil pencarian nilai *match heuristic*

Indeks	0	1	2	3	4
Pattern	N	A	N	D	A
MH	5	5	5	2	1

Contoh pencarian dengan algoritma *boyer-moore*

- Jika karakter yang dibanding tidak ada pada *pattern* maka nilai pergeserannya sepanjang jumlah *pattern*.
- Untuk pergeseran bandingkan nilai OH dan MH ambil nilai terbesar sebagai keputusan pergeseran.
- Jika semua karakter telah cocok, artinya *pattern* telah ditemukan dalam teks.

Tabel 2.7 Nilai Occurrence Heuristic dan match heuristic

Indeks	0	1	2	3	4
Pattern	N	A	N	D	A
MH	5	5	5	2	1
OH	4	3	2	1	0

Teks	N	A	N	D	O		D	A	N		N	A	N	D	A
Pattern	N	A	N	D	A										
Indeks	0	1	2	3	4										

Contoh implementasi pencarian pada teks proses ke-1

OH = O (Teks) =5

MH = index [4] (Pattern) = 1

5 > 1 dengan begitu pattern digeser 5 karakter

Teks	N	A	N	D	O		D	A	N		N	A	N	D	A
Pattern						N	A	N	D	A					
Indeks						0	1	2	3	4					

Contoh pencarian pada proses ke-2

OH = spasi = 5

MH = index [4] (Pattern) = 1

5 > 1 dengan begitu pattern digeser 5 karakter

Teks	N	A	N	D	O		D	A	N		N	A	N	D	A
Pattern											N	A	N	D	A
Indeks											0	1	2	3	4

Contoh pencarian pada proses ke-3

Semua karakter telah cocok, artinya *pattern* telah ditemukan di dalam teks.

3. HASIL DAN EKSPRIMEN

Pada proses algoritma *Boyer-Moore* dalam sistem yang dibuat adalah pada proses pencarian dengan algoritma Boyer-Moore dalam mencari kata. Pencarian algoritmanya sebagai berikut:

Indeks = 0 1 2 3 4 5

Pattern = D A N A U

MH = 5 5 5 2 1

OH = 4 3 2 1 0

OH = (length -1 -index) length = panjang karakter =5

“D” dengan Index = 0 **OH** = (5 - 1 - 0 = 4) maka nilai karakter “D” = 4

“A” dengan index = 1 **OH** = (5 - 1 - 1 = 3) maka nilai karakter “A” = 3

“N” dengan index = 2 **OH** = (5 - 1 - 2 = 2) maka nilai karakter “N” = 2

“A” dengan index = 3 **OH** = (5 - 1 - 3 = 1) maka nilai karakter “A” = 1

“U” dengan index = 4 **OH** = (5 - 1 - 4 = 0) maka nilai karakter “U” = 0

OH = (5 - 1 - 4 = 0) maka nilai karakter “A” = 0

Proses pencocokan

Teks = D A N A O

Pattern= D A N A U

Indeks = 0 1 2 3 4

Teks = D A N A U

Pattern= D A N A U

Indeks = 0 1 2 3 4

OH = O (Teks) =5

MH = index [4] (Pattern) = 1

5 > 1 dengan begitu pattern digeser 5 karakter

3.1 User Interface

Untuk mengimplementasikan sistem, aplikasi dibuat dengan menggunakan bahasa pemrograman *Java* dengan bantuan *Integrated Development Environment* yaitu *Android Studio* dan *Plugin Android Development Tools*. *User interface* dibuat dengan menggunakan fasilitas komponen visual yang disediakan oleh *Android Studio*.

3.2 Tampilan Awal

Tampilan awal program ini adalah tampilan dimana aplikasi pertama kali dijalankan. Adapun tampilan tersebut, sebagai berikut :



Gambar 4.7 Tampilan awal aplikasi Kamus.

Keterangan :

1. Button Indonesia-Arab berfungsi untuk menampilkan halaman pencarian Indonesia-Arab.
2. Button Arab-Indonesia berfungsi untuk menampilkan halaman pencarian Arab-Indonesia.
3. Button Tentang berfungsi untuk menampilkan info tentang pembuat aplikasi dan tujuannya.

3.3 Tampilan halaman Indonesia-Arab dan Arab-Indonesia

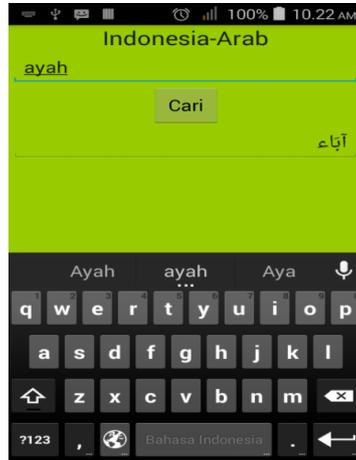
Halaman selanjutnya adalah halaman terjemahan Indonesia-Arab dan Arab-Indonesia, dimana saat user memilih tombol Indonesia-Arab atau Arab-Indonesia, maka akan langsung mengarah ke dalam halaman ini. Didalam halaman ini, terdapat satu buah tombol pencarian dan dua buah *textview* pada masing-masing halaman yang terdiri dari satu *text* untuk menerjemahkan kata dan satu *text* lagi untuk hasil terjemahan kata yang dicari.



Gambar 4.8 Halaman terjemahan Indonesia-Arab dan Arab Indonesia.

3.4 Tampilan hasil pencarian

Jika hasil pencarian user sesuai dengan data yang tersimpan di dalam sistem, maka akan muncul tampilan sebagai berikut :



Gambar 4.9 Tampilan Hasil Ditemukan.

Jika pencarian user tidak sesuai dengan data yang tersimpan di sistem, maka akan muncul tampilan sebagai berikut :



Gambar 4.10 Tampilan Hasil Tidak Ditemukan.

4. KESIMPULAN

Berdasarkan uraian diatas, mengenai Aplikasi Kamus Bahasa Indonesia-Arab Berbasis Android maka peneliti memberikan kesimpulan, sebagai berikut :

1. Aplikasi ini merupakan aplikasi sederhana yang membantu untuk menterjemahkan kata dalam bahasa Indonesia ke bahasa Arab dan sebaliknya.
2. Metode *Boyer-Moore* dapat diterapkan dalam pencocokan *String* untuk penyelesaian penterjemahan kata.

DAFTAR PUSTAKA

- Anny Yuniarti, 2012. *Membangun Aplikasi Kamus Bahasa Arab Online Sederhana*.
- Alfa, Eva, Satyaputra, 2014. *Beginning Android Pemrograming with ADT Budle*. Yogyakarta.
- Syarifah Aisyah, 2012. *Perancangan dan Pembuatan Aplikasi Kamus Bahasa Inggris-Indonesia-Aceh dan Aceh-Indonesia-Inggris Versi Mobile Berbasis Android*. Politeknik Lhokseumawe.
- Safaat H., Nazruddin. 2012. *Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika.
- Ramadhansyah, 2013. *Perancangan Aplikasi Kamus Bahasa Gayo dengan Menggunakan Metode Boyer-Moore*. *Jurnal Pelita Informatika Budi Darma IV(3):118-122*.
- Yudisti, Renditia. 2011. *Aplikasi Kamus Dwibahasa Indonesia - Inggris Berbasis Android*. *Jurnal*.