Penerapan Algoritma Pencarian BFS untuk Membentuk Prediksi *Pattern* pada Aplikasi Penulisan Huruf Arab

Alifa Nurani Putri (13511074)[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13511074@std.stei.itb.ac.id*
alifanuraniputri@gmail.com

***Abstrak*—Huruf Arab, dengan kedudukannya sebagai huruf penulisan bahasa Arab, merupakan salah satu huruf non-latin yang populer dan banyak digunakan di berbagai publikasi internasional. Seiring berkembangnya teknologi digital, banyak bermunculan aplikasi untuk mengetik huruf Arab, dimulai dengan aplikasi standar yang hanya mengganti huruf pada keyboard sampai aplikasi yang mampu melakukan transliterasi antar huruf. Salah satu jenis aplikasi yang *easy to use* adalah aplikasi yang memungkinkan pengguna untuk menuliskan input dengan huruf latin, lalu aplikasi akan memberikan prediksi sejumlah pattern yang mungkin dimaksud oleh pengguna. Makalah ini akan mencoba menganalisis salah satu metoda untuk membentuk *pattern-pattern* prediksi tersebut dengan algoritma *BFS*.**

***Kata kunci*—huruf arab, *BFS* , prediksi.**

I. PENDAHULUAN

Bahasa Arab merupakan salah satu bahasa tertua di dunia yang hingga kini termasuk bahasa populer dalam lingkup penggunaan internasional. Terlebih lagi bahasa Arab memiliki keunggulan sebagai bahasa yang digunakan dalam Kitab Suci *Al-Qur'an*, sehingga banyak sekali penggunaannya untuk berbagai publikasi seputar Agama Islam. Oleh karena itu, setelah memasuki era *digital* mulailah bermunculan berbagai cara untuk mengetikkan bahasa Arab. Cara yang paling klasik adalah hanya mengganti huruf pada *keyboard* dengan huruf arab. Dengan cara ini, banyak pengguna, terutama pemula yang merasakan kesulitan karena bahasa arab memiliki format yang tidak simpel seperti huruf latin. Banyak pengguna pemula memilih menuliskan maual dengan tangan. Cara ini dirasa kurang efketif dan tidak mengikuti perkembangan teknologi di era informasi ini.

Dari masalah yang telah diuraikan di atas, mulailah bermunculan beberapa aplikasi yang bertujuan mempermudah pengguna untuk mengetik huruf Arab. Aplikasi ini membidik sasaran pengguna pemula dengan intensitas penulisan tidak terlalu tinggi. Sistem kerja aplikasi ini adalah dengan memberikan prediksi *patern-patern* yang mungkin dimaksud pengguna, mengingat dengan satu *pattern* huruf latin bisa menjadi berbagai *pattern* dalam huruf Arab. Banyak cara untuk membentuk prediksi ini, dari yang sederhana dengan hanya melihat komposisi *pattern*, hingga dengan *heuristic* tertentu yang melibatkan daftar kosakata bahasa Arab.

Tujuan makalah ini adalah memberikan analisis mengenai salah satu cara pembentukkan prediksi *pattern* yang sederhana yaitu dengan membentuk *pattern-pattern* yang mungkin secara bertahap dari *sub-pattern*-nya dengan algoritma *BFS*.

II. TEORI DASAR

A. Huruf dan Penulisan Arab

Huruf Arab yang juga sering disebut huruf Hijaiyah merupakan salah satu bahasa tertua di dunia. Pada dasarnya, huruf arab terkomposisi dari konsonan-konsonan, dan vokal yang digunakan merujuk pada pemberian *harakat* pada konsonannya. Terdapat 3 *harakat* utama, yaitu :

- *Fathah* : baris yang merepresentasikan vokal 'a' dan pada beberapa huruf menyerupai vokal 'o'. Penandanya berupa garis pada bagian atas huruf.
- *Kasroh* : baris yang merepresentasikan vokal 'i'. Pendandanya berupa garis pada bagian bawah huruf.
- *Dommah* : baris yang merepresentasikan vokal 'u' penandanya berupa tanda *waw* (و) kecil di atas huruf.

Selain ketiga *harakat* utama tersebut, terdapat beberapa *harakat* lain: *Sukun* (mematikan huruf), *Tasydid* (penekanan karena huruf *double*), dan *Tanwin* (*harakat* yang bernilai sama dengan penambahan *nun* mati).

Huruf Arab yang digunakan saat ini sudah berevolusi dari bentuk awalnya. Evolusi ini meliputi penambahan titik untuk memperjelas perbedaan huruf dan menghindari ambiguitas, penambahan baris untuk mempermudah pembacaan terutama bagi orang *non*-arab, juga penambahan tanda panjang-pendek (kebanyakan hanya digunakan untuk pembacaan *Al-Qur'an* agar tidak terjadi

kesalahan pembacaan).

Pada dasarnya terdapat dua jenis penulisan Arab :

- Penullisan Arab Klasik : Penulisan Arab jenis ini digunakan dalam *Al-Qur'an* dan beberapa literatur klasik. Penulisan ini memiliki beberapa perbedaan gaya dan kosa kata dengan penulisan Arab modern.
- Penulisan Arab Standar Modern : Penulisan Arab Modern digunakan pada bahasa universal saat ini untuk berkomunikasi dalam bahasa Arab dan untuk penulisan pada buku, televisi, dan media lainnya.

Hal-hal lain yang menjadi ciri khas huruf dan penulisan Arab antar lain :

- Penulisan secara horizontal dari kanan ke kiri.
- Terdiri dari 28 huruf. Pada penulisan modern standar, terdapat beberapa tambahan huruf untuk menuliskan huruf-huruf yang tidak dapat terwakili secara tepat oleh 28 huruf utama, seperti 'p' dan 'g'.
- Tidak hanya digunakan untuk menulis bahasa Arab. Banyak bahasa lain yang di tulis dengan huruf Arab antara lain : *Azeri, Baluchi, Bosnian, Turkish.*
- Hampir semua huruf memiliki bentuk penulisan yang berbeda sesuai dengan peletakkannya pada suat kata (depan, tengah, atau belakang).

Gambar 2.1.1 28 + 1 *hamzah* Huruf Arab [5]

B. Transliterasi Arab-Latin

Transliterasi atau alih aksara merupakan penyalinan dengan pergantian huruf dari suatu abjad ke abjad yang lain. Transaliterasi biasa digunakan untuk kata-kata yang diserap atau diambil dari suatu bahasa dan ditulisakan ke dalam bahasa lain.

Pada Transliterasi Arab-Latin, terdapat beberapa pedoman yang dapat digunakan. Dari sejumlah pedoman transliterasi Arab-Latin, terdapat dua pedoman transliterasi yang sering digunakan bagi masyarakat Indonesia, yaitu :

- Transliterasi Qalam

Transliterasi Qalam biasa digunakan untuk melakukan transliterasi dari bahasa Arab ke bahasa Inggris.

- Transliterasi Kritis

Transliterasi Kritis biasa digunakan dalam transliterasi bahasa Arab ke bahasa Indonesia.

Berikut tabel yang memberikan gambaran perbedaan kedua trasliterasi tersebut :

Arab	Transliterasi Qalam (Inggris)	Transliterasi kritis (Indonesia)	Kata dari Transliterasi Qalam	Kata dari Transliterasi kritis
ـُ	O	u	**O**mar, **O**thman, **O**sama	**U**mar, **U**tsman, **U**samah
ث	Th	Ts	O**th**man, hadi**th**, Hadi**th**, Ibn Ka**th**ir, Ya**th**rib	U**ts**man, hadi**ts**, Haditsah, Ibnu Ka**ts**ir, Ya**ts**rib
ذ	Dh	dz	Abu **Dh**ar, Al-Tirmi**dh**i	Abu **Dz**ar, At-Tirmi**dz**i
ش	Sh	Sy	Ai**sh**a, Qurai**sh**, **Sh**ihab, **Sh**ia	Ai**sy**ah, Qurai**sy**, **Sy**ihab, Syi'ah
ص	S	Sh	**S**ahih	**sh**ahih
ظ	Z	Zh	al-Hafi**z**	al-Hafi**zh**
ة	t, h (luluh dalam penyerapan)	t, h	Abraha, Aqaba, Amina, Aisha, Alqama, fitna, Haditha, Shia, sura, Osama	Abraha**h**, Aqaba**h**, Aisya**h**, Alqama**h**, fitna**h**, Haditsa**h**, Syi'a**h**, sura**h**, Usama**h**

Tabel 2.2.1 Transliterasi Qalam & Kritis [3]

C. Aplikasi Penulisan Huruf Arab

Sejak dimulainya era teknologi berbasis komputer, mulailah bermunculan cara-cara untuk menuliskan huruf Arab di komputer. Hingga kini, banyak sekali bermunculan aplikasi yang digunakan untuk mengetik huruf Arab. Semakin maju, aplikasi penulisan huruf Arab pun semakin memudahkan pengguna dengan menambah sejumlah fitur yang membantu pengguna dalam menuliskan huruf arab.

Berikut adalah beberapa cara untuk menulis huruf Arab, dari yang kalsik hingga yang didukung oleh aplikasi modern:

- **Menggunakan pengaturan khusus agar *keyboard* mengeluarkan input huruf arab pada komputer.**

Cara ini merupakan salah satu cara paling klasik yang digunakan untuk menulis huruf Arab. Untuk menjalankannya cukup mengubah *setting*-an

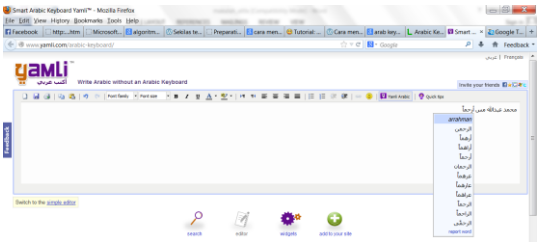
pada *control panel* komputer, lalu kita dapat mengetikkan huruf arab pada aplikasi pengolah kata layaknya *Ms. Word*. Cara ini dianggap cukup sulit, terutama untuk pengguna pemula yang hanya membutuhkan penulisan huruf Arab dengan intensitas rendah, karena pengguna harus terlebih dahulu terbiasa/hafal letak huruf yang bersesuaian pada *keyboard*, jika tidak pengguna harus memunculkan *virtual-keyboard*, dan menjadikannya acuan selama penulisan.

- **Menggunakan Aplikasi Transliterasi tanpa *Hint*.**

Gambar 2.3.1 *Screenshot* aplikasi transliterasi tanpa *hint* [7]

Aplikasi ini memungkinkan kita untuk menulis huruf arab dengan huruf latin, lalu aplikasi tersebut akan melakukan transliterasi secara otomatis. Kelemahan aplikasi ini yaitu pengguna sulit untuk menentukan *pattern* huruf pada suatu kata sesuai keinginannya, sering terjadi kesalahan dari maksud yang diinginkan pengguna.

- **Menggunakan Aplikasi Transliterasi dengan *hint* prediksi *pattern*.**



Gambar 2.3.2 *Screenshot* aplikasi transliterasi dengan *hint* prediksi [4]

Pada umumnya, prinsip kerja aplikasi ini sama dengan poin aplikais pada poin sebelumnya, hanya aplikasi ini menambah fitur khusus berupa prediksi *pattern*, mengingat suatu kata dalam huruf latin dapat dituliskan dalam berbagai jenis *pattern* dalam huruf Arab. Dengan aplikasi ini, pengguna bisa mendapatkan hasil yang sesuai dengan kebutuhannya. Aplikasi ini sangat cocok untuk pengguna pemula yang menggunakan aplikasi penulisan huruf Arab dalam intensitas yang tidak terlalu besar.

Dari sejumlah cara tersebut, aplikasi jenis ke-tiga mempunyai keunggulan lebih karena dapat memudahkan pengguna untuk menulis sesuai harapannya. Berikut adalah ilustrasi perbedaan hasil yang bisa didapat dari jenis aplikasi ke-dua dan ke-tiga. Terlihat bahwa aplikasi ke-tiga memberikan hasil lebih baik.

Teks yang ditulis : "Muhammad Abdullah Musa Arrahman"

موهاممماد ابدوللاه موسا اررااهمان

Gambar 2.3.3 Hasil transliterasi otomatis aplikasi jenis ke-dua [7]

Gambar 2.3.4 Hasil transliterasi dengan memilih hint pada aplikasi jenis ke-tiga [4]

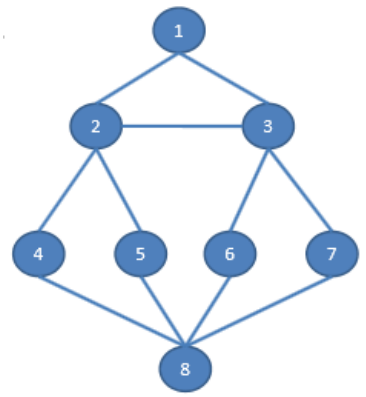
D. Algoritma Pencarian

Algoritma pencarian merupakan salah satu algoritma yang sering digunakan untuk menyelesaikan permasalahan. Untuk meningkatkan kemangkusan pencarian, telah ditemukan berbagai jenis algoritma pencarian yang masing-masing mempunyai kekurangan dan kelebihan. Fokus utamanya adalah penggunaan algoritma yang mangkus dalam artian cepat dan hemat memori untuk menyelesaikan suatu permasalahan.

Pada makalah ini akan dibahas penemuan solusi suatu permasalahan yang pencariannya berbasis pembangunan pohon. Berikut adalah beberapa algoritma pencarian populer yang pencariannya berbasis pada pembangunan pohon pencarian :

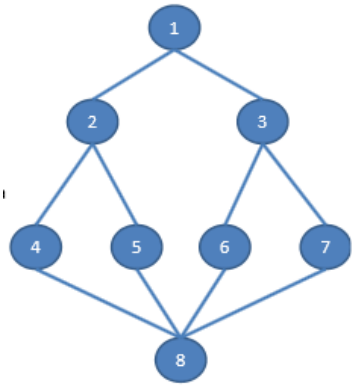
- Algoritma *BFS (Breadth First Search)*

Algoritma BFS (akan dijelaskan secara spesifik pada bagian E) merupakan algoritma pencarian tanpa informasi yang diketahui, sehingga pembangunan pohon dilakukan semaksimal mungkin untuk mencari kemungkinan ditemukannya solusi. Algoritma BFS merupakan algoritma dengan mencarian melebar. Secara umum, langkahnya adalah dengan memulai mengunjungi suatu simpul pada suatu level pohon, lalu mengunjungi setiap simpul yang bertetangga dengan simpul yang telah dikunjungi, secara traversal pada stiap level pohon. Berikut adalah gambaran sketsa pencarian BFS.



Gambar 2.4.1 Pencarian BFS [1]

•Algoritma *DFS (Depth First Search)*
Layaknya algoritma BFS, algoritma DFS juga pmerupakan algoritma pencrian tanpa inoformasi. Perbedaannya adalah pada simpul yang dikunjunginya. Pada algoritma DFS, pencarian dilakukan mendalam dari suatu simpul, hingga ditemukan solusi atau hingga batas yang ditentukan. Jika solusi belum ditemukan dan telah melampaui batas, makan akan berpindah ke simpul yang lain (tetangga). Cara kerjanya dimulai dengan pertama mencari semua anak pohon yang mungkin lalu memlusuri salah satu dan terus berulang seperti itu. Berikut adalah sketsa pencarian DFS :



Gambar 2.4.2 Pencarian DFS [1]

D. Cara Kerja Algoritma Pencarian BFS

Seperti yang telah digambarkan secara ringkas sebelumnya, Algoritma BFS adalah algoritma pencarian berbasis pembangunan pohon secara melebar. Dimana pada setiap levelnya, setiap simpul pada level tersebut akan ditelusuri kemungkinan-kemungkinannya. Pengunjungan simpul dilakukan secara *pre-Order* yaitu mengunjunsi suatu simpul dalam satu level terlebih dahulu, lalu mengunjungi simpul-simpul lain yang bertetengga dengan simpul tersebut terlebih dahulu. Jadi dalam pohon yang dibentuknya, semua simpul pada level *d* akan dikunjungi telebih dahulu dari simpul-simpul pada level *d+1*.

Proses untuk mengunjungi setiap simpul pada pohon dengan pola di atas dapat diwujudkan dengan menerapkan sistem antrian atau *queue*. Setiap simpul yang dikunjungi akan langsung masuk ke *queue* sebagai acuan untuk mengunjungi tetangga-tetangganya kemudian. Tiap simpul hnaya masuk ke dalam antrian tepat sekali. Hal lain yang dibutuhkan adalah nilai *boolean* yang menyatakan apakah suatu simpul pada antrian sudah dikunjungi atau belum, adar tidak ada simpul yang dikunjungi lebih dari sekali.

Secara ringkas, berikut adalah struktur data yang diperlukan untuk membuat sebuah pencarian BFS:

1. Matriks ketetanggaan A = [a_{ij}] yang bernilai satu jika i dan j bertetangga dan bernilai 0 jika i dan j tidak bertetangga.
2. Antrian *q* untuk menyimpan simpul yang telah dikunjungi.
3. Tabel boolean yang menyatakan apakah simpul telah dikunjungi
   `dikunjungi : array [1..n] of boolean`
   `dikunjungi[i]` bernilai true jika sudah dikunjungi
   `dikunjungi[i]` bernilai false jika belum dikunjungi.

Berikut adalah langkah-langakah dari algoritma pencarian BFS :

1. Masukkan simpul akar (*start*) ke dalam antrian.
2. Ambil simpul pada awal antrian dan periksa laha apakah simpul tersebut merupakan solusi.
3. Jika simpul merupakan solusi, pencarian selesai, hasil dikembalikan.
4. Jika simpul bukan solusi, telusuri kemungkinan selanjutnya, masukkan seluruh simpul yang bertetangga (simpul anak) ke dalam antrian.
5. Jika antrian kosong dan setiap simpul sudah diperiksa, pencarian selesai dan hasil tidak ditemukan.
6. Ulangi pencarian untuk setiap simpul dalam antrian dari no 2.

III. PEMBAHASAN: PEMBENTUKAN PREDIKSI PATTERN PENULISAN HURUF ARAB DENGAN MENGGUNAKAN ALGORITMA BFS

A. Gambaran Umum Cara Kerja Algoritma

Pada algoritma ini, akan dicari sejumlah *pattern* yang mungkin dibentuk dengan huruf Arab dari suatu masukan kata dalam huruf latin. Algoritma ini hanya mencari *pattern* yang bersesuaian secara *exact* (100% kesamaan) transliterasinya dengan masukan. Sehingga tidak terdapat keberagaman panjang pendek dan keberagaman huruf yang bisa diakibatkan dari kesalahan maksud penulisan masukan huruf latin. Diasumsikan input yang dimasukkan penggunasesuai dengan pedoman transliterasinya.

Ide dasar dari algoritma ini adalah mencari sebanyak mungkin kombinasi *pattern* dengan 'memainkan' *substring-substring* dari masukan kata. Pembentukannya dilakukan dengan pembangunan pohon secara BFS, dimana pada setiap simpul terdapat suatu *substring* yang akan dikonversi ke huruf Arab. Jadi Proses konversi dilakukan pada setiap simpul. Pencarian berakhir saat pohon telah mencapai semua kemungkinan yang ada. Atau bisa saja pencarian dibatasi pada level tertentu jika pohon terlaluu besar.

Pada setiap simpul, akan dicari simpul selanjutnya yang merupakan *substring-substirng* setelahnya dari masukan huruf latin, dengan *range* 1-3 huruf. Pemilihan *range substring* dari 1-3 dikarenakan suatu huruf dengan harakat dalam bahasa arab mungkin bisa dituliskan dalam 1-3 huruf latin, dikarenakan ada huruf sukun (satu huruf latin) dan huruf bertasydid (dua huruf latin) beserta harakat barisnya. Pada program ini, karena hanya menganalisis per *pattern*, terdapat penanganan kasus khusus, level 1 pohon tidak mungkin terdiri dari satu huruf konsonan karena tidak dapat dilafalkan.

Dari pohon yang terbentuk, suatu prediksi didapatkan dari setiap rute yang mungkin dari akar ke daun. Mungkin saja pada beberapa simpul ditemukan simpul dengan *substring* yanag tidak mungkin ditulis dalam huruf Arab. Simpul seperti itu akan berakhir penelusurannya, karena tidak mungkin ditemukan solusi prediksi pada anak pohonnya.

B. Format Masukan dan Pedoman Transliterasi

Masukan ditulis dalam hurf latin dengan ketentuan sebagai berikut :

- Huruf Ganda Konsonan merepresentasikan huruf ber*tasydid* atau suatu huruf sukun dan dilanjutkan pola lain.
- Huruf Ganda Vokal merepresentasikan huruf panjang, perbedaan panjang pembacaan antara 2 atau >2 *harakat* tidak ditangani pada algoritma ini. Setiap huruf masukan yang terdeteksi bagian dari jenis ini akan ditangani sebagai kasus khusus, dapat dikonversi menjadi bentuk huruf bergaris atas. Hal ini dilakukan agar *range substring* cukup dari 1-3, tidak perlu 1-4 pada pengangan kasus huruf bertasydid dan panjang.

Adapun, pedoman Transliterasi yang digunakan mengikuti transliterasi diplomatis, yang juga sering digunakan di Indonesia, dengan tabel sebagai berikut :

Huruf Arab	Alih aksara	Keterangan
ا		
ب	B b	
ت	T t	
ث	Ts ts	
ج	J j	
ح	Ḥ ḥ	h dengan satu titik di bawah
خ	Kh kh	
د	D d	
ذ	dz dz	
ر	R r	
ز	Z z	
س	S s	
ش	Sy sy	
ص	Sh Sh	
ض	Dl dl	
ط	Th th	
ظ	Zh zh	
ع	‘	voiced pharyngeal fricative
غ	Gh gh	
ف	F f	
ق	Q q	
ك	K k	
ل	L l	
م	M m	
ن	N n	
ه	H h	
و	W w	
ء	‘	
ي	Y y	

Tabel 3.2.1 Pedoman Transliterasi (diambil dari Pedoman Transliterasi Diplomatis) [3]

C. Pseudo Code Algoritma

```
function isCanTransliterate(input s:
string -> boolean) {

//mengembalikan true jika s bisa di
transliterasikan dari hurfu latin ke
huruf Arab
//algoritma diasumsikan sudah ada, bukan
menjadi fokus makalah ini

if (s bisa di transliterasikan)
     return true
else
     return false
}

----------------------------------------

//Queue

procedure buatQueue(input/output
q:Queue){
//membuat antrian kosong
}

procedure masukkanQueue(input/output
q:Queue, input s: string){
//meemasukkan s ke dalam q pada posisi
belakang
}

procedure hapusQueue(input/output
q:Queue, input s: string){
//menghapus s dari kepala q
}

function isAntrianKosong (input q:Queue
-> boolean){
//mengembalikan true jika antrian kosong
dan false jika masih ada antrian
}

----------------------------------------

//Tree

procedure buatTree(input/output t:Tree,
input s : string){
//memembuat Tree baru dengan akar s
}

procedure setAsChild(input/output t:Tree,
input n:Simpul s : string){
//menjadikan s anakn dari Simpul n,
sekaligus memasukkan anak ke dalam
antrian Queue
}

procedure setSimpulBuntu(input/output
t:Tree, input n:Simpul){
//mengeset simpul n buntu, sehingga
simpul tersebut tidak diproses lagi
}

procedure setPatternResults(input/output
t:Tree, output p : array of Pattern ){
//menndapatkan sejumlah pattern hasil
pencarian prediksi dari pohon BFS yang
sudah selesai terbentuk , dengan setiap
pattern diambil dari setiap rute yang
mungkin dibuat dan tidak buntu dari akar
ke daun
}
```

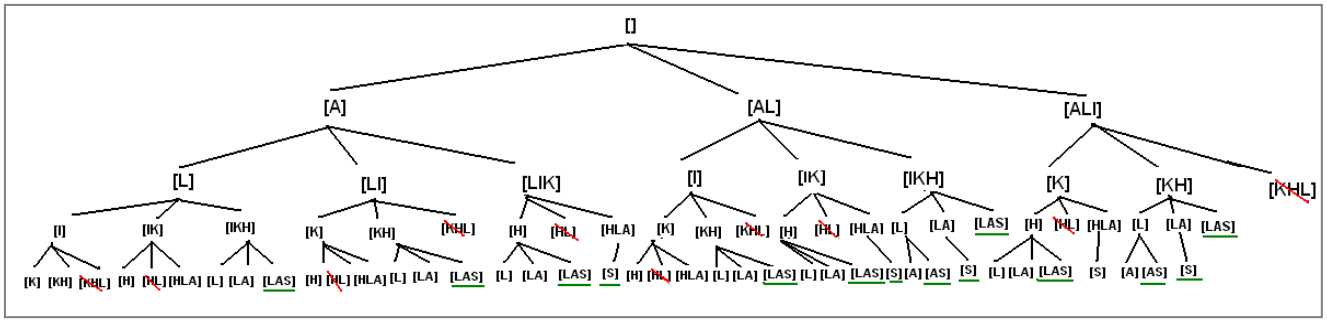
```
procedure
buildBFSPredictionForArabicWriting(input
s: string) {
//membentuk prediksi-prediksi pattern
untuk penulisan bahasa Arab dengan
transliterasi

Tree<String> t;
Queue<String> q;
Array of pattern p; //result prediksi
untuk diubah ke abjad arab

start = string kosong;

buatQueue(q);
masukkanQueue(q,start);
buatTree(start);
dikunjungi[start]=true; //start telah
dikunjungi

while (isAntrianKosong==0) do {
     hapusQueue(q,start); //hapus simpul
     yang sudah dikunjungi

     for (setiap simpul n yang
     bertetangga dengan start) {

          if (dikunjungi[n]==false) {
               masukkanQueue(q,n);
               dikunjungi[n] = true;

               array [3] of string anak;
               //membuat substring-
               substring simpul anak

               total_length_akar_ke_n;
               //terdefinisi sebagai
               jumlah huruf yang sudah
               dipakai dari akar ke simpul
               n.

               for (i=3;i>0;i--) {
               //mengeset seluruh
               kemungkinan anak dari
               simpul, anak yang mungkin
               berups substring lanjutan
               1-3 huruf

               //anak dgn jumlah huruf i
               if (s.length-
               total_legnth_akar_ke_n>(i-
               1)) {
                    anak[i-1]=
                    s.substr(0,i);
                    setAsChild(t,n,anak[i-
                    1];)
                    if
                    (isCanTransliterate(an
                    ak[i-1])==false) {
                         setBuntu(simpul
                         anak[i-1]);
                    //simpul di-set buntu
                    jika tidak bisa
                    ditransliterasi
                    }
               }
               }
          }
     }
}
//antrian kosong, pohon selesai dibentuk
setPatternResult(t,p)
//pattern-pattern prediksi terbuentuk
}
```

D. Contoh Pohon Pencarian Pattern Prediksi

Dari penelusuran algoritma tersebut diharapkan terbangun pohon solusi yang terbangun per-level. Pada gambar di bawah ini, dilakukan pencarian prediksi untuk kata "*alikhlas*". Gambar tersebut merupakan pohon penelusuran presiksi pattern yang mungkin dari kata "*alikhlas*" sampai pada level 4 pohon (hanya untuk ilustrasi pencarian, karena jika dilanjutkan, ukuran pohon menjadi sangat besar untuk disketsakan) :

Keterangan Pohon :

1. Garis Hijau Bawah : Menujukkan daun, atau penemuan suatu *pattern* dari kata.
2. Garis Diagonal Merah : *pattern* terebut tidak dapat ditransliterasikan.



Gambar 3.4.1 Pohon (level 0-4) Penelusuran Presdiksi kata "alikhlas" (gambar dibuat dengan aplikasi paint)

Sampai pada level 4 pohon tersebut, telah ditemukan beberapa *pattern* prediksi yang mungkin (digambarkan dengan garis bawah hijau), yaitu :

(terurut sesuai *pattern* yang lebih dulu ditemukan dalam pencarian BFS)

1. [AL] – [IKH] – [LAS]
2. [ALI] – [KH] – [LAS]
3. [A] – [L] – [IKH] – [LAS]
4. [A] – [LI] – [KH] – [LAS]
5. [A] – [LIK] – [H] – [LAS]
6. [A] – [LIK] – [HLA] – [S]
7. [AL] – [I] – [KH] – [LAS]
8. [AL] – [IK] – [H] – [LAS]
9. [AL] – [IK] – [HLA] –[S]
10. [AL] – [IKH] – [L] – [AS]
11. [AL] – [IKH] – [LA] – [S]
12. [ALI] – [K] – [H] – [LAS]
13. [ALI] – [KH] – [L] – [AS]
14. [ALI] – [KH] – [LA] – [S]

Dari sejumlah pattern yang ditemukan sampai pada level 4 pohon di atas, Saat ditulis dalam huruf Arab (pada makalah ini tidak dituliskan langsung dalam huruf Arab), ditemukan beberapa *pattern* sama, yang pada list di atas direpresentasikan dengan warna yang sama, maka hasil dapat dikelompokkan menjadi seperti berikut :

1. [AL] – [IKH] – [LAS]
   [A] – [L] – [IKH] – [LAS]
   [AL] – [IKH] – [LA] – [S]
   [AL] – [I] – [KH] – [LAS]
2. [ALI] – [KH] – [LAS]
   [A] – [LI] – [KH] – [LAS]
   [ALI] – [KH] – [LA] – [S]
3. [A] – [LIK] – [H] – [LAS]
   [A] – [LIK] – [HLA] – [S]
   [ALI] – [K] – [H] – [LAS]
4. [AL] – [IK] – [H] – [LAS]
   [AL] – [IK] – [HLA] –[S]
5. [AL] – [IKH] – [L] – [AS]
6. [ALI] – [KH] – [L] – [AS]

Sehingga dapat ditemukan bahwa dengan metode pencarian *pattern* dengan BFS untuk kata "*alikhlas*" dengan penelusuran pohon tidak lengkap, berakhir pada level 4, ditemukan 6 *pattern* yang penurlisan arabnya berbeda. Jika pencarian dilanjutkan, mungkin saja ditemukan *pattern* baru yang berbeda, namun bisa juga *pattern-pattern* yang ditemukan selanjutnya hanya merupakan bentuk *redundant* dari ke-6 hasil di atas.

IV. Analisis Hasil dan Saran Perbaikan

Dari contoh pada bab V, terbukti bahwa algoritma pencarian BFS dapat membentuk sejumlah *pattern* berbeda jika ditulis dalam huruf Arab.

Keunggulan dari penggunaan metoda pencarian BFS untuk kasus ini antara lain :

- Dibanding dengan algoritma DFS, pemakaian algoritma BFS lebih cepat menemukan prediksi, karena *pattern* prediksi bisa diemukan pada level-level bawah pohon.
- Dapat menemukan sebanyak mungkin *pattern* prediksi, tidak berhenti saat ditemukan satu solusi.

Disamping keunggulan, pemakaian metoda murni pencarian BFS untuk mendapatkan *pattern-pattern*

prediksi juga memiliki beberapa kelemahan :

- Karena pencarian ini meupakan pencarian tanpa basis pengetahuan, banyak ditemukan *pattern* prediksi yang aneh dan sebenarnya tidak pernah ditemukan dalam bahasa Arab.
- Algoritma yang digunakan pada makalah ini belum dapat menangani huruf-huruf yang transliterasinya mirip kerena mungkin pengguna menggunakan transliterasi yang berbeda.

Dari hasil dan analisis di atas, terdapat beberapa saran pengembangan ynag bisa dilakukan untuk mengembangkan sebuah *engine* yang dapat memberikan prediksi-prediksi *pattern* pada transliterasi Latin-Arab, yaitu :

- Mengkombinasikan metoda pencarian BFS dengan menambahkan basis pengetahuan beruap sejumlah kata yang biasa digunakan dalam bahasa Arab. Sehingga *engine* tetap dapat menampilkan banyak variasi *pattern* sebanyak mungkin, namun dapat menampilkan hasil lebih baik dengan menempatkan kata-kata yang telah terdaftar di basis pengetahuan lebih atas dibading kata yang lain.
- Menangani kasus kesalahan transliterasi dari sisi pengguna, dengan mengeluarkan prediksi dari huruf yang mirip. Misalnya huruh ‘Tsa‘ , pada beberapa transliterasi ditulis dengan huruf ‘Tha’.
- Menangani kasus keberagaman tanda panjang, ada baiknya jika pengguna cukup memasukkan katanya tanpa perlu kebingungan menuliskan bentuk pajang-pendeknya, lalu *engine*-lah yang akan menampilkan beberapa variasi panjang pendek yang mungkin dibentuk dari kata masukan. Hal ini akan lebih memudahkan user dalam pengetikan input, dan membuat hasil yang muncul lebih akurat.

V. KESIMPULAN

Algoritma pencarian BFS (*Breadth First Search*) dapat digunakan untuk membentuk prediksi *pattern* dari transliterasi Latin-Arab pada aplikasi penulisan huruf Arab yang mendukung fitur *hint* prediksi. Dengan metoda ini, didapatkan variasi *pattern* dengan jumlah banyak dan *pattern* solusi dapat ditemukan dalam waktu singkat. Adapun metoda ini mempunyai kelemahan utama yaitu sejumlah *pattern* yang terbentuk bisa saja tidak pernah ada dalam bahasa Arab.

VI. ACKNOWLEDGMENT

Makalah ini dibuat unuk kuliah IF2211 Strategi Algoritma 2013/2014. Penulis hendak mengucapkan terima kasih kepada kedua Dosen mata kuliah IF 2211, yaitu Ibu Masayu Leylia Khodra selaku dosen kelas penulis (K2), dan Bapak Rinaldi Munir selaku dosen kelas K1, atas ilmu yang telah disampaikan Bapak dan Ibu selama keberlangsungan mata kuliah ini. Selain itu, penulis memohon maaf atas kekurangan-kekurangan makalah ini antara lain keterbatasan penulis untuk secara langsung melakukan analisis hasil dengan huruf Arab.

REFERENSI

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/ Slide Bahan Kuliah Strategi Algoritma 2013/2014, 19 Desember 2013, 02:02 WIB.
[2] http://www.omniglot.com/writing/arabic.htm , 18 Desember 2013 00:43 WIB.
[3] http://jogjaare.blogspot.com/2012_12_01_archive.html , 18 Desember 2013 01:56 WIB
[4] http://www.yamli.com/arabic-keyboard/ , 19 Desember 2013, 01:08 WIB.
[5] http://1.bp.blogspot.com/-UwdY_pBx8o0/T-Bri9vdnqI/AAAAAAAAAT4/Xkn6-wOoNLE/s1600/Graphic1.jpg , 18 Desember 2013, 01:34 WIB.
[6] http://onbuble.wordpress.com/2011/05/26/6/ , 19 Desember 2013, 00:37 WIB.
[7] http://www.lexilogos.com/keyboard/arabic.htm , 19 Desember 2013 01:24 WIB.

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, 20 Desember 2013

Alifa Nurani Putri
13511074