

# Penggunaan Algoritma Boyer Moore Untuk Pencarian Arsip Multimedia Pada Perangkat Lunak Pemutar Media

Archie Anugrah - 13508001  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
e-mail: if18001@students.if.itb.ac.id

## ABSTRAK

Dalam makalah ini, penulis membahas algoritma pencarian menggunakan algoritma pencocokan *string* yang dapat diaplikasikan dalam melakukan pencarian pada perangkat lunak pemutar media. Kekhasan dari pencarian pada perangkat lunak pemutar media adalah dalam setiap pencarian, selain memeriksa nama arsip, perangkat lunak juga harus memeriksa isi dari *metadata* masing-masing arsip multimedia. Pemeriksaan pada setiap arsip dalam *playlist* membuat penentuan algoritma pencocokan *string* yang tepat menjadi penting untuk menentukan performa dari algoritma pencarian secara umum. Dengan makalah ini, diharapkan pembaca dapat mengerti standar berpikir dalam membentuk algoritma pencarian terhadap arsip multimedia.

**Kata kunci:** Pencarian pada perangkat lunak pemutar media, pencocokan *string*, dan *metadata*.

## 1. PENDAHULUAN

### 1.1 Pendahuluan

Sejak kompresi arsip audio dan video dapat dilakukan, banyak pemakai komputer yang mengoleksi arsip multimedia digital pada komputer mereka. Seiring dengan perkembangan kapasitas *harddisk* yang cepat, semakin hari, makin banyak pula koleksi rata-rata arsip multimedia yang dimiliki oleh pemakai pada umumnya. Banyaknya arsip multimedia dan ketersebarannya membuat pemakai sering merasa kesulitan mencari arsip yang diinginkan. Karena hal tersebut, muncullah fitur pencari yang dilakukan terhadap *playlist* yang sedang digunakan. Dengan fitur pencari ini, pemakai dapat menemukan arsip yang diinginkan dengan mudah dan cepat. Pada saat ini, fitur pencari merupakan salah satu fitur yang penting dan wajib ada pada perangkat lunak pemutar media.

### 1.2 Pemutar Media

Pemutar media merupakan istilah umum untuk mengacu kepada perangkat lunak komputer yang dapat

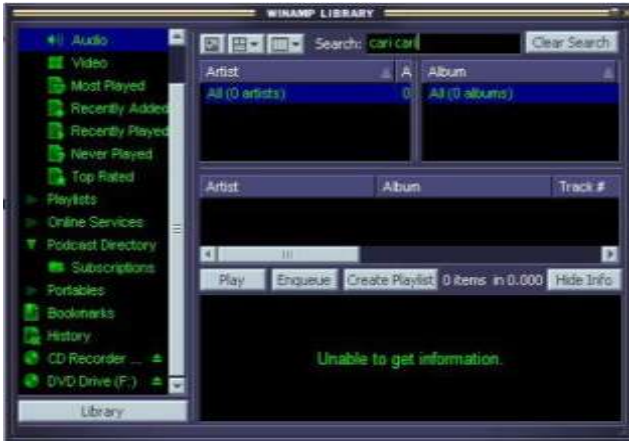
memainkan berkas multimedia<sup>1</sup>. Pemutar media memiliki berbagai macam tipe, mulai dari pemutar media *audio*, *video*, ataupun keduanya. Beberapa pemutar media yang umum digunakan meliputi Windows Media Player, Winamp, VLC, MPlayer, Xine, dan Totem.

Dalam perangkat lunak pemutar media, biasanya terdapat fitur pencarian arsip multimedia berdasarkan kata kunci tertentu. Pencarian biasanya dilakukan terhadap semua komponen *metadata* ataupun nama dari arsip multimedia tertentu. Jika arsip yang dicari ditemukan, arsip tersebut akan disorot oleh perangkat lunak.

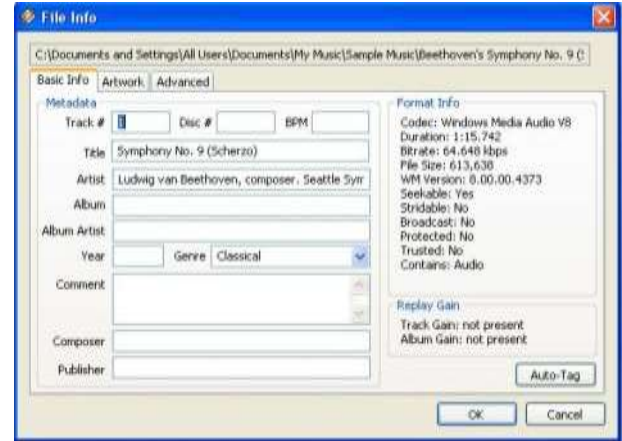


Gambar 1. Winamp Media Player, salah satu perangkat lunak pemutar media

<sup>1</sup> [http://id.wikipedia.org/wiki/Pemutar\\_media](http://id.wikipedia.org/wiki/Pemutar_media)



Gambar 2. Fitur pencarian pada Winamp Media Player



Gambar 3. ID3 tag, metadata arsip MP3

### 1.3 String Matching

Menurut Wikipedia, *Pattern Matching* adalah “act of checking some sequence of tokens for the presence of the constituents of some pattern”<sup>2</sup>. *String matching* adalah salah satu tipe dari *pattern matching*, hanya saja dalam *string matching*, objek yang dibandingkan adalah *string*. Dalam proses ini, sebuah *string* diperiksa apakah *string* tersebut terdapat pula di *string* yang lainnya. Ada banyak algoritma mengenai *string matching*, salah satunya adalah *brute force*, KMP, dan Boyer Moore.

### 1.4 Arsip multimedia

Arsip multimedia adalah arsip pada komputer yang berhubungan dengan multimedia, baik itu audio ataupun video. Dalam arsip multimedia, umumnya terdapat *metadata* yang memberikan informasi tentang arsip yang bersangkutan.

### 1.5 Metadata

*Metadata* adalah data yang berfungsi untuk menerangkan data yang lain. *Metadata* digunakan pada banyak aplikasi, bukan hanya untuk multimedia saja. Dalam arsip multimedia, terdapat *metadata* yang umumnya berisi judul, artis, tahun rilis, dan *genre* arsip multimedia yang bersangkutan. *Metadata* yang terdapat dalam arsip multimedia juga beragam, contoh yang paling populer adalah ID3 tag, *metadata* yang digunakan oleh arsip MP3. Pada awal pengembangan, *metadata* dari arsip multimedia sering tidak digunakan, tetapi seiring perkembangan arsip yang semakin banyak, saat ini *metadata* menjadi komponen yang penting untuk menyediakan informasi mengenai arsip multimedia yang bersangkutan.

## 2. METODE

### 2.1 Algoritma pencarian *string*

Arsip multimedia memiliki beberapa sifat, yaitu, nama arsip pendek, memiliki banyak komponen *metadata*, dan setiap komponen *metadata* memiliki isi *string* yang relatif pendek. Walaupun arsip multimedia memiliki banyak komponen *metadata*, tetapi pada umumnya yang dicari hanyalah komponen judul, artis, tahun, dan genre karena selain umum ada pada setiap arsip multimedia, komponen-komponen tersebut juga merupakan komponen *metadata* yang paling penting.

Ada beberapa cara perbandingan *string* yang memungkinkan untuk digunakan. Yang pertama adalah algoritma *brute force*. Dalam algoritma ini setiap karakter dicocokkan satu persatu. Algoritma ini tidak efektif, sehingga sebaiknya algoritma ini tidak digunakan. Algoritma yang memungkinkan lainnya adalah KMP dan Boyer Moore. Sebenarnya secara umum, kedua algoritma ini mirip dalam penanganan perbandingan. Setiap karakter dibandingkan, tetapi jika ada karakter yang tidak sama, loncatan yang terjadi lebih besar dibandingkan *brute force* sehingga jumlah perbandingan pun menjadi lebih sedikit, hanya saja Boyer Moore membandingkan karakter dari belakang sedangkan KMP membandingkan dari depan. Dalam hal performa, performansi kedua algoritma ini mirip dalam menangani *string* yang pendek, sehingga manapun algoritma yang dipilih, hasilnya tidak akan jauh berbeda. Dalam makalah ini, algoritma yang dipilih untuk digunakan adalah Boyer Moore karena walaupun untuk *string* yang pendek, KMP dan Boyer Moore memiliki performansi yang mirip, tetapi Boyer Moore lebih umum untuk digunakan.

### 2.2 Boyer Moore

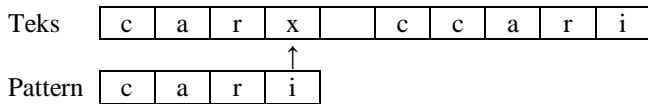
Dari sekian banyak metoda *string matching*, Boyer Moore adalah algoritma yang memiliki performa yang

<sup>2</sup> [http://en.wikipedia.org/wiki/Pattern\\_matching](http://en.wikipedia.org/wiki/Pattern_matching)

baik dan umum digunakan oleh perangkat lunak dengan fitur pencari kata. Algoritma Boyer Moore memiliki ciri khas, yaitu perbandingan dilakukan dari belakang dan dapat menentukan berapa jauh lompatan harus dilakukan berdasarkan perbandingan karakter yang tidak cocok.

Langkah-langkah pencarian Boyer Moore dapat dijabarkan sebagai berikut :

- Periksa kecocokan pattern dengan teks, karakter demi karakter, dimulai dari karakter paling belakang.

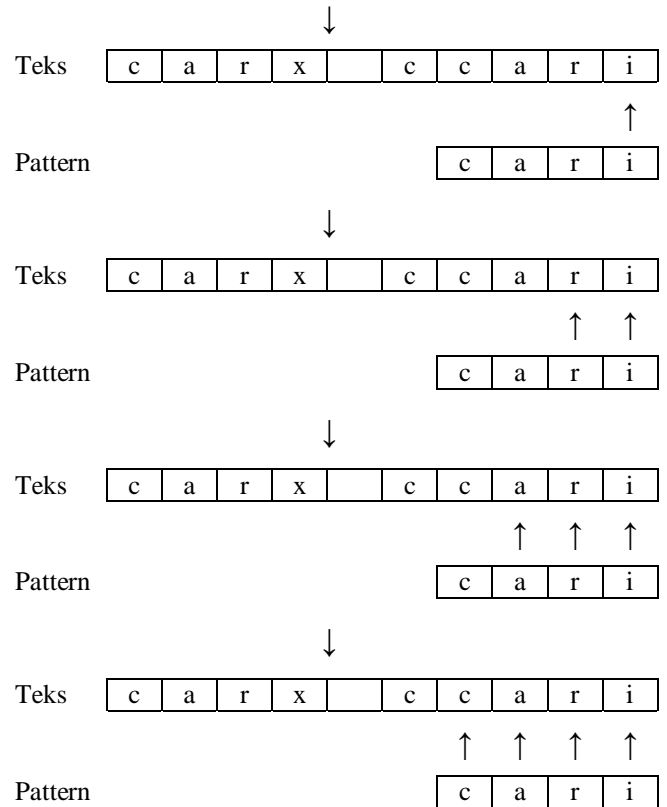
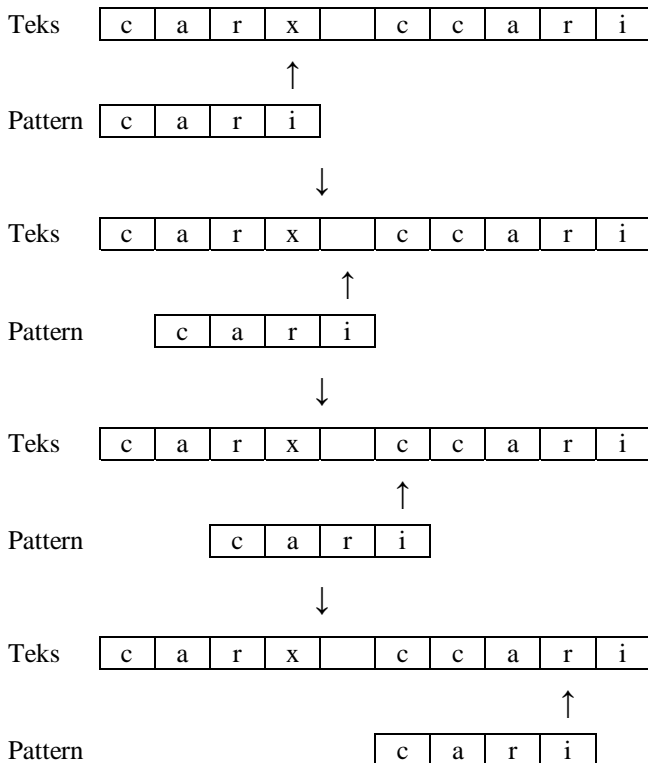


- Jika ada karakter yang tidak sama, maka dilakukan pelompatan pengecekan.

- Jika pada pattern ada karakter x, maka geser pattern sehingga kedua x bertemu.
- Jika pada pattern ada karakter x, tetapi pergeseran tidak dapat membuat kedua karakter x itu bertemu, maka geser pattern 1 langkah.
- Pada kasus selain kedua kasus diatas, geser pattern 1 langkah.

- Untuk mempercepat pengecekan karakter x tersebut, maka sebelum melakukan komputasi, tabel lokasi kemunculan masing-masing karakter dihitung terlebih dahulu.

Visualisasi pencarian teks diatas dengan menggunakan Boyer Moore adalah sebagai berikut :



Berikut adalah potongan fungsi Boyer Moore dalam bahasa Java <sup>3</sup> :

```
public static int[] buildLast(String pattern)
/* Return array storing index of last
occurrence of each ASCII char in pattern. */
{
    int last[] = new int[128]; // ASCII char set

    for(int i=0; i < 128; i++)
        last[i] = -1; // initialize array

    for (int i = 0; i < pattern.length(); i++)
        last[pattern.charAt(i)] = i;

    return last;
} // end of buildLast()

public static int bmMatch(String text, String pattern)
{
    int last[] = buildLast(pattern);
    int n = text.length();
    int m = pattern.length();
    int i = m-1;

    if (i > n-1)
```

<sup>3</sup><http://fivedots.coe.psu.ac.th/Software.coe/LAB/PatMatch/PatternMatching.ppt>

```

return -1; // no match if pattern is
// longer than text

int j = m-1;
do {
if (pattern.charAt(j) == text.charAt(i))
if (j == 0)
return i; // match
else { // looking-glass technique
i--;
j--;
}
else { // character jump technique
int lo = last[text.charAt(i)]; //last occ
i = i + m - Math.min(j, 1+lo);
j = m - 1;
}
} while (i <= n-1);

return -1; // no match
} // end of bmMatch()

```

### 2.3. Algoritma Pencarian Arsip Multimedia

Dalam melakukan pencarian, perangkat lunak pemutar media pada umumnya hanya mencari arsip multimedia yang terdapat dalam *playlist* perangkat lunak bersangkutan. Dalam setiap arsip multimedia, nama arsip fisik dan *metadata* harus diperiksa dan dicocokkan dengan *pattern* yang dimasukkan oleh pengguna. Algoritma ini, akan memeriksa semua arsip dan mengeluarkan angka nomor *playlist* arsip yang ditemukan dan mengeluarkan -1 jika data tidak ditemukan. Setelah arsip ditemukan, indeks yang dikeluarkan dari fungsi dapat digunakan untuk menyorot *playlist*.

Pseudocode :

**Function** CariPlaylist (input playlist : array of arsipmultimedia, input pattern : string) → integer

#### **Kamus**

i : integer {digunakan untuk mencacah playlist}

{Fungsi pencarian Boyer Moore}

**Function** bmMatch(input text : String, input pattern : String) → Integer

{Fungsi yang mengembalikan *string* isi dari tag dengan tipe field tertentu dan mengembalikan *string* kosong jika tipe field tersebut tidak terdefinisi dalam arsip multimedia}

**Function** GetTag(input arsip : arsipmultimedia, input TipeField : string) → string

{Fungsi yang mengembalikan *string* nama arsip dari arsipmultimedia}

**Function** GetArsipname(input arsip : arsipmultimedia) → string

#### **Algoritma**

i ← 0 {inisialisasi pencacah}

**While** (i < playlist.size) **do** {iterasi isi playlist}

**If**

((bmMatch(GetTag(playlist[i], "Title"), pattern) ≠ -1)

or

(bmMatch(GetTag (playlist[i], "Artist"), pattern) ≠ 1)

or

(bmMatch(GetTag (playlist[i], "Year"), pattern) ≠ 1)

or

(bmMatch(GetTag (playlist[i], "Genre"), pattern) ≠ 1)

or

(bmMatch(GetArsipname(playlist[i]), pattern) ≠ 1)) **then**

→ i {jika tepat, return index}

**endif**

i = i+1

**endwhile**

→ -1 {jika tidak tepat, return -1}

### 3. KESIMPULAN

Dalam perangkat lunak yang berbasis pemutar media, pencarian arsip merupakan fitur yang penting. Pencarian arsip dapat dilakukan dengan memanfaatkan algoritma Boyer Moore yang dilakukan pada nama arsip dan isi seluruh komponen *metadata* dari arsip tersebut. Dalam perbandingan string perangkat lunak pemutar media, algoritma Boyer Moore merupakan algoritma yang cukup efektif dan akurat.

### REFERENSI

- [1] Munir, Rinaldi. 2009. Diktat Kuliah IF3051 Strategi Algoritma. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- [2] [http://id.wikipedia.org/wiki/Pemutar\\_media](http://id.wikipedia.org/wiki/Pemutar_media) (Waktu akses : 15 November 2010, 16:55 PM)
- [3] [http://en.wikipedia.org/wiki/Pattern\\_matching](http://en.wikipedia.org/wiki/Pattern_matching) (Waktu akses : 15 November 2010, 17:22 PM)
- [4] <http://en.wikipedia.org/wiki/ID3> (Waktu akses : 17 November 2010, 14:30 AM)
- [5] <http://fivedots.coe.psu.ac.th/Software.coe/LAB/PatMatch/PatMatch.ppt> (Waktu akses : 17 November 2010, 14:54 AM)

## **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, 21 November 2010

A handwritten signature in black ink that reads "Archie". The signature is written in a cursive style and is underlined with a single horizontal line.

Archie Anugrah  
13508001