

# Implementasi Algoritma Boyer-Moore pada *Automatic TV Series Downloader*

Karunia Ramadhan 13508056  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
if18056@students.if.itb.ac.id

**Abstrak**—Hiburan multimedia merupakan salah satu hal yang cukup penting bagi manusia. Salah satu jenis hiburan multimedia yang ada adalah serial TV. Makalah ini akan membahas bagaimana mengenali dan mengunduh serial TV inputan secara langsung dari sebuah portal *bitTorrent* dengan perbandingan *string* menggunakan algoritma Boyer-Moore. *File* multimedia tersebut nantinya akan diunduh dengan protokol *bitTorrent* secara otomatis dengan program yang ditentukan.

**Kata Kunci**—Algoritma Boyer-Moore, Hiburan multimedia, Protokol *BitTorrent*, Serial TV.

## I. PENDAHULUAN

Multimedia adalah media dan konten yang menggunakan kombinasi dari berbagai bentuk konten yang berbeda. Kombinasi ini termasuk teks, audio, gambar, animasi, video, dan konten interaktif. Multimedia mungkin tidak sepenting perkembangan dan teknologi lain dalam dunia IT, tetapi kebutuhan akan multimedia juga akan selalu berkembang sepanjang waktu. Andrew S. Tanenbaum, pembuat buku *Operating System* dan *Computer Networks* yang menjadi pegangan beberapa kuliah kami mahasiswa informatika ITB, bahkan mendedikasikan bab khusus untuk membahas multimedia dari berbagai segi, penanda bahwa multimedia merupakan bidang yang cukup penting. Konten multimedia sendiri biasanya direkan dan dimainkan, ditambahkan dan diakses dengan alat pemroses konten informasi seperti komputer dan alat elektronik lain seperti TV.

Multimedia bisa dibagi menjadi dua bagian besar, linear dan non-linear. Linear berarti konten akan berjalan tanpa navigasi dari penonton dan non-linear berarti ada interaksi antar pengguna dengan media (interaktif). Salah satu contoh konten multimedia linear adalah serial TV.

Serial pada TV adalah serial yang menitik-beratkan pada kontinuitas plot yang akan dibuka dalam episode-episode yang sekuensial. Serial ini biasanya mengikuti *arc* cerita utama dalam satu musim serial maupun pada seluruh *series*-nya, membedakannya dari serial tv

tradisional yang lebih episodik dan menitik-beratkan pada episode yang terpisah-pisah.

Acara-acara ini menuntun para pemirsanya untuk selalu mengikuti tiap episode untuk mengetahui ceritanya, dan penemuan alat perekam seperti VCR dan DVR telah mempermudah para penonton untuk selalu mengikuti serial yang ditontonnya. Di internet sendiri, banyak sekali komunitas penggemar serial TV yang selalu memperbaharui basis data tontonannya tiap musim dan menyebarkan *file* multimedia dari serial TV hasil rekaman kepada masyarakat luas.

Dalam satu musim serial TV, biasanya banyak sekali serial yang beredar secara bersamaan dan keluar setiap minggu. Hal ini menyebabkan sang pengguna yang ingin menontonnya harus selalu *menongkrong* portal-portal dimana setiap hari serial tersebut keluar supaya bisa mengunduh dan selalu menonton tiap episodenya secara *up-to-date*. Bisa dibayangkan banyak sekali waktu yang terbuang apabila penonton mengikuti banyak serial sekaligus, meskipun waktu ini akan berkurang bila kita sudah mengenali jadwal keluar serial tersebut.

Untuk mengatasi masalah borosnya waktu tersebut, penulis menggunakan sebuah metode yang memanfaatkan teknologi RSS, protokol *BitTorrent*, dan algoritma pencocokan *string* Boyer-Moore. Pengguna akan bisa menentukan serial apa saja yang ingin dia tonton pada musim itu, lalu aplikasi akan secara otomatis mencocokkan input serial tersebut dengan RSS *update* dari portal. Aplikasi kemudian akan memperbaharui basis data dan mengunduh *torrent* episode terbaru dari serial tersebut. *File torrent* itu sendiri kemudian akan dimanfaatkan oleh program aplikasi *torrent* untuk mengunduh *file* multimedia episode tersebut.

## II. DASAR TEORI

### A. Really Simple Syndication (RSS)

RSS adalah salah satu keluarga dari format web *feed* yang digunakan untuk mem-*publish* pekerjaan yang sudah diperbaharui, seperti entri blog, *headline* dari sebuah berita, munculnya audio dan video baru, dan lain

lain dalam sebuah format yang sudah distandarisasi. Sebuah dokumen atau *feed* dari TSS mengandung ringkasan teks dari pembaharuan tersebut dan metadata seperti tanggal *publish* dan penulis artikel tersebut. RSS *feed* ini bisa dibaca menggunakan perangkat lunak (agregator dsb.) berbagai jenis seperti *web-based*, *desktop-based*, maupun *mobile-based*.

Format RSS dispesifikasikan menggunakan XML, spesifikasi generik pada pembuatan format data. Meskipun format RSS sudah berubah jauh dari awalnya saat Maret 1999, baru pada 2005-2006 RSS digunakan secara luas. [1] Berikut ini adalah contoh dari sebuah RSS *feed* :

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <title>RSS Title</title>
  <description>This is an example of an RSS
feed</description>
  <link>http://www.someexemplerssdomain.co
m/main.html</link>
  <lastBuildDate>Mon, 06 Sep 2010 00:01:00
+0000 </lastBuildDate>
  <pubDate>Mon, 06 Sep 2009 16:45:00
+0000 </pubDate>
  <item>
    <title>Example entry</title>
    <description>Here is some text containing an
interesting description of the thing to be
described.</description>
    <link>http://www.wikipedia.org/</link>
    <guid>unique string per item</guid>
    <pubDate>Mon, 06 Sep 2009 16:45:00
+0000 </pubDate>
  </item>
</channel>
</rss>
```

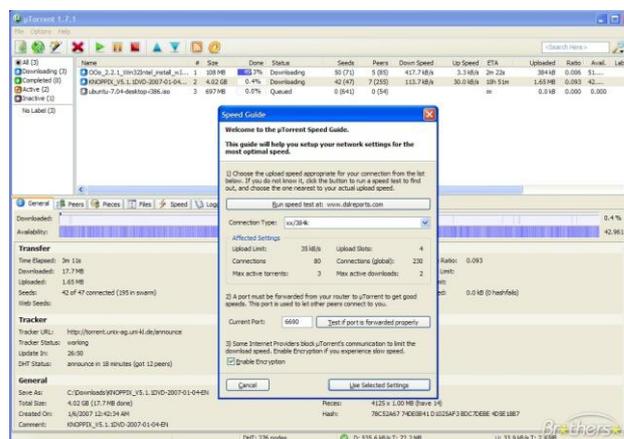
### B. BitTorrent Protocol

BitTorrent adalah sebuah protokol *peer-to-peer file sharing* yang digunakan untuk mendistribusikan data yang besar. BitTorrent adalah salah satu protokol yang paling sering digunakan untuk mentransfer data, dengan estimasi 27% sampai 55% dari semua trafik pada internet dipakai oleh protokol ini (Februari 2009). [2]

Protokol BitTorrent bisa digunakan untuk mendistribusikan data tanpa menggunakan banyak sumber daya pada komputer *source* maupun jaringan. Daripada mengunduh *file* dari satu *source*, protokol ini membuat pengguna untuk bergabung pada suatu kumpulan dari para *host* untuk mengunduh dan

mengunggah dari satu sama lain secara simultan. Pengguna yang ingin mengunggah sebuah *file* sebelumnya akan membuat *file torrent* yang akan didistribusikan dengan cara konvensional di internet. Dia kemudian membuat *file* tersebut bisa diunduh melalui *BitTorrent node* dan berperilaku sebagai *seed*. Orang lain yang memiliki *file torrent* bisa mengambil *file* tersebut kepada simpul *BitTorrent* mereka, berperilaku sebagai *peer* atau *leecher*, mengunduh dengan membuat koneksi kepada *seed* maupun *peer* lain.

*File* yang didistribusikan itu sendiri dibagi menjadi segmen-segmen yang disebut *pieces*. Ketika setiap *peer* menerima *piece* baru dari *file*, dia akan menjadi sumber *piece* tersebut kepada *peer* lain. Dengan *BitTorrent*, tugas untuk mendistribusikan *file* dibagi ke semua orang yang mau memiliki *file* tersebut.



Contoh Gambar Program BitTorrent : uTorrent

### C. Algoritma Boyer-Moore

Pada persoalan pencocokkan *string*, akan diberikan sebuah teks berisi *n* karakter dan *pattern* (teks yang akan dicari) berisi *m* karakter dengan *m* < *n*. Dengan algoritma-algoritma yang ada, akan dicari lokasi pertama dimana *pattern* tersebut muncul dalam teks karakter dimana pencarian dilakukan.

Algoritma pencarian *string* Boyer-Moore adalah algoritma pencarian yang cukup berbeda dengan algoritma lainnya, dimana pada algoritma ini pencarian *pattern* dilakukan secara terbalik dari belakang ke depan *pattern*. Misalkan kita mencari sebuah kata "KARU" pada sebuah teks, maka pencarian akan dimulai dari awal teks dan akan mengecek huruf ke-4 apakah dia mengandung huruf "U". Bila ya, dia akan mundur ke huruf ke-3 dan mengecek apakah dia mengandung huruf "R" dan begitu seterusnya sampai dia mengecek huruf di posisi pertama dengan "K".

Boyer-Moore menggunakan pendekatan terbalik ini karena kasus yang akan terjadi bila ada kesalahan pada verifikasi karakter. Misal bila huruf "U" tidak ditemukan pada posisi ke-4 dan kita malah menemukan huruf "X", dimana disini "X" tidak ada pada *pattern* "KARU" yang berarti tidak ada sama sekali karakter yang cocok dari

karakter pertama pada teks sampai karakter ke-4 yang mengikutinya. Setelah verifikasi, kita akan bisa maju dan mengecek kembali pada karakter ke-8 untuk mencari huruf "U". Hal ini menyebabkan biasanya semakin panjang *pattern* yang dicari, semakin cepat juga algoritma ini menemukan hasilnya.

```

- - - - - X - - - - -
A N P A N M A N - - - - -
- A N P A N M A N - - - - -
- - A N P A N M A N - - - - -
- - - A N P A N M A N - - - - -
- - - - A N P A N M A N - - - - -
- - - - - A N P A N M A N - - - - -
- - - - - - A N P A N M A N - - - - -
- - - - - - - A N P A N M A N - - - - -
- - - - - - - - A N P A N M A N - - - - -

```

**Ilustrasi pengecekan *string* dari belakang**

Sebelum pencarian dilakukan, algoritma akan menghitung dua tabel yang akan digunakan untuk mendapatkan informasi setiap ada. Tabel pertama berisi informasi berapa posisi dimana pencarian selanjutnya akan dilakukan berdasarkan karakter yang menyebabkan kesalahan verifikasi (*Bad Characters Shift Table*) dan tabel kedua berisi informasi yang sama berdasarkan berapa banyak karakter yang benar sama sebelum verifikasi gagal dalam satu pencarian *pattern* (*Good Suffix Shift Table*). Algoritma akan maju ketika verifikasi karakter gagal sesuai nilai yang paling besar dalam dua tabel tersebut.

**III. IMPLEMENTASI PENYELESAIAN MASALAH**

**A. Konsep**

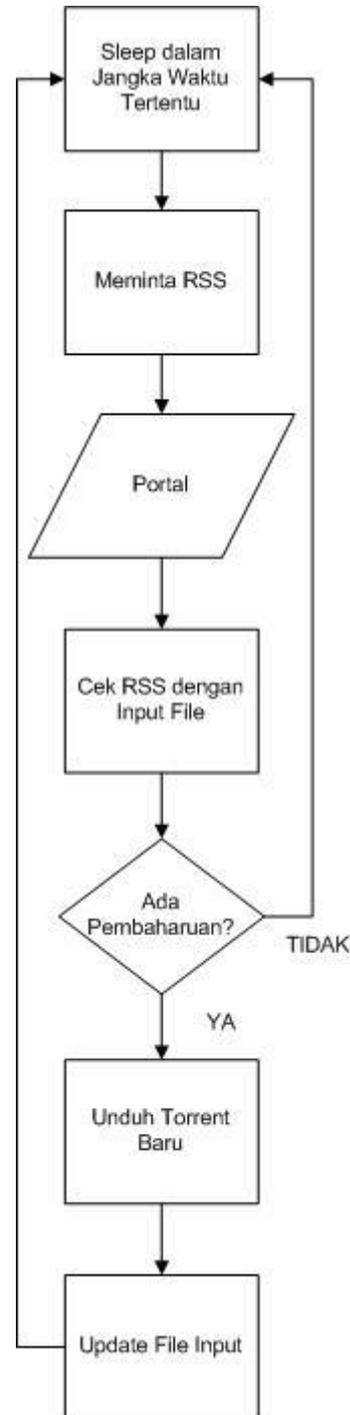
Konsep penyelesaian masalah *automatic downloader* ini sendiri cukup sederhana :

- Definisikan serial-serial tv yang diikuti dengan format tertentu pada *file* teks, contoh : [nama serial] [musim] [episode terakhir].
- Rekues RSS *feed* dari situs portal (*HTTP Request*).
- Simpan data RSS dari *update* terakhir sampai yang paling baru. Untuk penggunaan pertama kali, ambil *feed* sampai panjang yang sudah didefinisikan (contoh : 50 buah).
- Bandingkan semua argumen input pada *file* teks dengan data RSS dengan algoritma *Boyer-Moore*. Simpan *link* bila perbandingan tersebut cocok.
- Unduh semua *file torrent* dari *link* yang sudah ditemukan dan masukkan dalam suatu *folder*.
- Perbaharui *file* teks sesuai unduhan.
- Gunakan protokol *BitTorrent* untuk mengunduh

semua *file* yang *torrent*-nya ada di folder tersebut. Pindahkan *file torrent* bila *file* sudah terunduh.

- Ulangi rekues setiap jangka waktu tertentu.

Berikut ini adalah diagram sederhana yang menggambarkan bagaimana konsep cara kerja dari aplikasi *automatic downloader* ini :



**Ilustrasi Konsep Cara Kerja Aplikasi**

## B. Implementasi

Aplikasi ini belum sempat diimplementasikan, tetapi harusnya cukup mudah untuk dibuat dalam bahasa tingkat tinggi yang memiliki banyak API atau *library* yang cukup lengkap seperti C# maupun Java. Aplikasi ini kurang lebih hanya melakukan HTTP Request untuk mendapatkan RSS *feed* lalu membandingkannya dengan *string* masukan dari *file* eksternal. Pencocokkan *string* disini juga bisa menggunakan algoritma lain seperti KMP maupun *brute-force* biasa. Penulis sendiri menggunakan algoritma Boyer-Moore karena penulis merupakan penggemar algoritma tersebut.

Implementasi algoritma Boyer-Moore sendiri kurang lebih sama seperti Tugas Besar 3 IF13051 yang terdiri dari dua tahap, *preprocessing* tabel di awal dan pencarian. Pada tahap pertama, akan dihitung nilai dari tabel *Bad Characters* dan *Good Suffix*. Kemudian pada tahap pencarian, program akan mencari *pattern* pada teks dengan verifikasi dan maju sesuai kesalahan yang terjadi dan nilai dari tabel.

Tabel *Bad Characters* dihitung dari *string pattern* dengan cara memasukkan posisi tertinggi semua karakter alfabet pada *pattern*. Bila *pattern* tidak mengandung karakter tersebut maka nilainya adalah -1, yang nanti akan menyebabkan pergeseran sebesar panjang *pattern* tersebut.

Tabel *Good Suffix* dihitung dari *pattern* yang ada pada karakter saat verifikasi salah. Dua kasus yang terjadi adalah *pattern* yang cocok terjadi juga di *string pattern* dan hanya sebagian dari *pattern* yang cocok ada pada awal dari *string pattern*. Dari kedua kasus tersebut nilai yang paling besar diambil untuk setiap *pattern* karakter dari *i* sampai panjang *string pattern*.

Implementasi Boyer-Moore dalam fungsi akan berjalan seperti berikut :

- Hitung panjang *pattern* dan teks.
- Bila panjang *pattern* 0, kembalikan teks. Bila panjang teks 0, kembalikan teks. Bila tidak keduanya, lanjutkan.
- Hitung Tabel *Bad Characters Shift* dan *Good Suffix Shift*.
- Loop dari awal teks sampai posisi panjang teks – panjang argumen,
- Inisiasi variabel panjang argumen, misalkan *j*.
- Bila  $j > 0$  (*pattern* belum semua sama) dan karakter pada posisi  $j-1$  pada *pattern* dengan karakter yang sedang dibandingkan sama, kurangi *j* (bandingkan karakter sebelumnya).
- Kalau hanya  $j > 0$  (sebuah karakter tidak ditemukan), bandingkan nilai pada kedua tabel untuk mencari *shift* yang paling besar. Maju sesuai nilai tersebut.
- Bila  $j = 0$  (*pattern* ditemukan), kembalikan teks yang sudah dipotong untuk ditampilkan (*assign return variable and break*).
- Ulangi sampai semua teks dicari.

Bila ditemukan *string* yang cocok, aplikasi akan menyimpan nama, informasi, dan URL dalam sebuah struktur data. Struktur data ini nantinya akan digunakan ketika aplikasi akan mengunduh *file torrent* karena yang akan diunduh hanyalah serial TV yang ingin ditonton (sudah didefinisikan dan ditemukan *update* episode terbarunya dari situs portal).

Untuk mengunduh *file* serial TV sendiri dari *file torrent*, bisa digunakan berbagai macam aplikasi *BitTorrent* seperti uTorrent. Setelah *file torrent* terkumpul di suatu *folder*, aplikasi cukup memperbaharui *file* teks input dan membiarkan *file* diunduh dengan program eksternal tersebut. Kira-kira pada implementasi nanti, program uTorrent sudah memiliki konfigurasi bahwa dia akan mengunduh menggunakan *folder* yang sudah didefinisikan (*folder* tempat kita menaruh *torrent file* hasil unduhan aplikasi) dan akan memindahkan *torrent file* maupun menghapusnya bila *file* sudah terunduh. Dengan selalu melakukan HTTP Request dalam jangka waktu tertentu, sekarang pengguna tidak perlu lagi mengawasi portal secara manual karena *file* serial TV yang diinginkan sudah akan diunduh secara otomatis begitu keluar.

## IV. KESIMPULAN

Untuk memudahkan para penonton serial TV yang mengunduh serial favoritnya dari internet untuk tidak membuang waktu mengecek portal untuk memperbaharui serial, dapat digunakan teknik pencocokkan *string* dan konsep RSS untuk mengunduh serial secara otomatis. Implementasi aplikasi akan memanfaatkan HTTP Request dan membandingkan *string* dengan data RSS yang sudah terformat untuk mengetahui *update* dari suatu serial. Perbandingan ini akan berlangsung setiap jangka waktu tertentu, dan jika perbandingan berhasil (yang berarti telah muncul episode baru dari serial yang pengguna ikuti), aplikasi akan mengunduh *file torrent* sesuai hasil perbandingan. Pengunduhan *file* multimedia sebenarnya akan diurus oleh protokol *BitTorrent*. Aplikasi hanya akan melempar *file torrent* ke suatu *folder* dan pengaturan *torrent file* tersebut selanjutnya akan dilakukan oleh program klien *BitTorrent* seperti uTorrent yang sudah dikonfigurasi. Dengan penyelesaian masalah seperti ini, pengguna tidak perlu lagi mengecek situs portal secara manual dan proses pengunduhan serial favorit pengguna tiap minggu akan dilaksanakan secara otomatis.

## DAFTAR REFERENSI

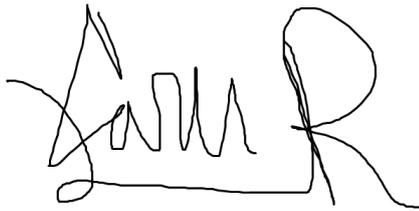
- [1] "Icons: It's still orange". Microsoft RSS Blog. December 14, 2005. Diambil pada 2008-11-09.
- [2] Internet Study 2008/2009.*ipoque* (Leipzig, DE).

## 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 2010

ttd

A handwritten signature in black ink, appearing to read 'Karunia R', with a large, stylized 'R' at the end.

Karunia Ramadhan 13508056