

Pendeteksian Plagiarisme Musik dengan Algoritma Boyer-Moore

Nicholas Rio - 13510024¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹nicholasrio2811@gmail.com

Abstract—Tren penjualan musik saat ini telah bergeser dari manual dan analog menjadi penjualan digital. Di satu sisi perubahan ini baik, namun di sisi lain perubahan ini menyebabkan maraknya plagiarisme musik. Oleh karena itu, untuk mendeteksi plagiarisme musik dapat digunakan aplikasi lebih lanjut dari Algoritma Boyer-Moore yang telah dipelajari selama pelajaran Strategi Algoritma. Solusi yang terdapat pada karya tulis ini pun masih memungkinkan untuk dikembangkan lebih jauh, tidak terbatas pada musik tertulis saja.

Index Terms—Boyer-Moore, Musik, Pattern Matching, Tokenisasi.

I. PENDAHULUAN

Pada zaman modern ini, industri musik mulai berganti tren dari paper based dan analog based, menjadi digital based. Pada satu sisi, perubahan tren ini adalah sesuatu yang positif. Mengapa? Karena dengan perubahan menuju tren yang baru ini, distribusi musik (baik partitur maupun lagunya secara langsung) dapat dilakukan secara jauh lebih efisien. Namun hal ini mendatangkan masalah baru. Selayaknya barang-barang digital lainnya seperti software dan game, musik pun tak luput dari pembajakan. Pembajakan yang penulis maksudkan di sini bukan hanya pengkopian file musik secara ilegal, namun juga pembuatan musik yang mirip dengan musik yang telah ada sebelumnya. Kejadian ini lebih dikenal dengan sebutan plagiarisme musik.

Oleh karena itu, penulis tertarik untuk mengembangkan salah satu algoritma pattern-matching yang telah dipelajari, yaitu algoritma Boyer-Moore. Pengembangan algoritma ini dibuat untuk mendeteksi apakah suatu bagian dari musik merupakan suatu bagian dari musik lain yang dibandingkan bersama.

Pada makalah ini, musik yang dibandingkan akan terbatas pada musik-musik yang tertulis (bukan audio file). Namun kemungkinan pengembangan algoritma agar dapat mendeteksi plagiarisme musik pada audio file akan dibahas juga oleh penulis pada subbab-subbab berikutnya.

II. ALGORITMA BOYER-MOORE

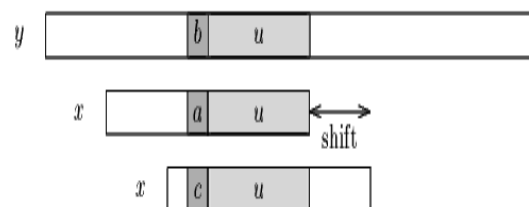
Algoritma Boyer-Moore merupakan salah satu algoritma pattern-matching yang terkenal. Algoritma ini dipublikasikan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Perbedaan utama algoritma ini dari pattern-matching biasa adalah pengecekannya yang dimulai dari kanan dan bukan dari kiri. Algoritma Boyer-Moore diketahui sebagai algoritma pattern-matching yang memiliki performa paling baik dalam pencarian kata berbasis bahasa Inggris.

Algoritma Boyer-Moore melakukan proses awal pada suatu pattern P dengan cara membuat boundary function terlebih dahulu. Boundary function dari pattern P berisi hal-hal berikut :

- Karakter apa saja yang terdapat dalam pattern P
- Last occurrence dari suatu karakter yang terdapat dalam pattern P.

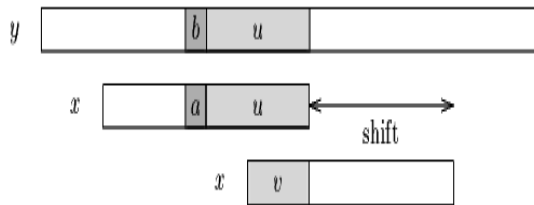
Setelah boundary function dibuat, proses berlanjut dengan perbandingan karakter dari pattern P terhadap string S yang hendak dibandingkan, dari kanan ke kiri.

Cara kerja algoritma ini adalah sebagai berikut :



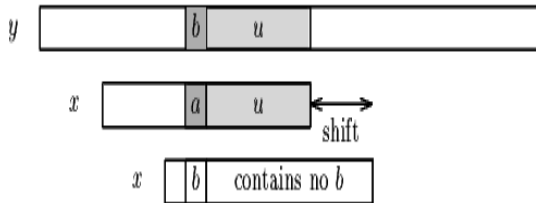
Gambar 1

Pada tipe pergeseran yang pertama ini, u kembali terdapat di pattern x setelah bergeser.



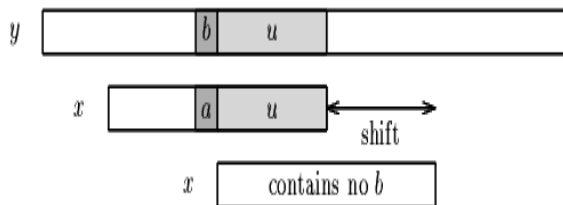
Gambar 2

Pada tipe pergeseran yang kedua, sebagian suffix dari u terdapat kembali sebagai prefix pattern x.



Gambar 3

Pada tipe pergeseran yang ketiga, tidak terdapat u di dalam pattern x yang digeser. Namun, karakter b yang terakhir dibandingkan pada string y masih ditemukan pada pattern x.



Gambar 4

Pada tipe pergeseran yang keempat, karakter terakhir yang ditemukan pada string y tidak terdapat pada pattern x. Oleh karena itu, pattern x di-shift seluruhnya ke kanan sedemikian rupa sehingga bagian paling kiri dari pattern berada di sebelah kanan karakter terakhir string y yang dibandingkan dengan pattern x.

Algoritma Boyer-Moore memiliki kelebihan terutama ketika jumlah karakter yang mungkin terdapat pada string maupun pattern berjumlah banyak. Kompleksitas dari algoritma ini adalah $O(m \cdot n)$ untuk kasus terburuk, namun hal ini jarang terjadi jika jumlah karakter yang tersedia cukup banyak.

III. MUSICXML

MusicXML adalah suatu cara untuk menuliskan not balok menjadi suatu Extensible Markup Language biasa. Format penulisan ini pertama kali ditemukan oleh Recoradre LLC pada tahun 2004. Tujuan utama dari penulisan dalam bahasa MusicXML adalah untuk mempermudah pertukaran partitur yang ditulis diantara program-program scorewriter (penulis not balok).

Program-program populer yang telah mengadopsi format MusicXML ini antara lain : MakeMusic Finale, Avid Sibelius, MuseScore, Capella.

Sebagai contoh dari penulisan dalam format MusicXML, mari kita lihat gambar berikut ini :



Gambar 5

Gambar di atas adalah contoh penulisan not balok yang benar. Bila direpresentasikan menjadi format MusicXML, hasilnya adalah demikian (penulis hanya mengutip baris pertama tanpa formatting) :

```
<divisions>256</divisions>
<note color="#000000" default-x="57">
  <pitch>
    <step>G</step>
    <octave>5</octave>
  </pitch>
  <duration>512</duration>
  <instrument id="P1-I1" />
  <voice>1</voice>
  <type>half</type>
  <stem>down</stem>
  <staff>1</staff>
</note>
<note color="#000000" default-x="165">
  <pitch>
    <step>E</step>
    <octave>5</octave>
  </pitch>
  <duration>256</duration>
  <instrument id="P1-I1" />
  <voice>1</voice>
  <type>quarter</type>
  <stem>down</stem>
  <staff>1</staff>
</note>
<note color="#000000" default-x="228">
  <pitch>
    <step>F</step>
    <octave>5</octave>
  </pitch>
  <duration>256</duration>
  <instrument id="P1-I1" />
  <voice>1</voice>
  <type>quarter</type>
  <stem>down</stem>
  <staff>1</staff>
</note>
</measure>
<measure number="2" width="230">
  <note color="#000000" default-x="15">
    <pitch>
      <step>G</step>
      <octave>5</octave>
    </pitch>
    <duration>512</duration>
    <instrument id="P1-I1" />
    <voice>1</voice>
    <type>half</type>
    <stem>down</stem>
    <staff>1</staff>
  </note>
  <note color="#000000" default-x="122">
    <pitch>
      <step>C</step>
      <octave>5</octave>
    </pitch>
    <duration>512</duration>
    <instrument id="P1-I1" />
    <voice>1</voice>
    <type>half</type>
```

```

<stem>down</stem>
<staff>1</staff>
</note>
</measure>
<measure number="3" width="269">
<note color="#000000" default-x="15">
<pitch>
<step>D</step>
<octave>5</octave>
</pitch>
<duration>256</duration>
<instrument id="P1-I1" />
<voice>1</voice>
<type>quarter</type>
<stem>down</stem>
<staff>1</staff>
</note>
<note color="#000000" default-x="78">
<pitch>
<step>C</step>
<octave>5</octave>
</pitch>
<duration>256</duration>
<instrument id="P1-I1" />
<voice>1</voice>
<type>quarter</type>
<stem>down</stem>
<staff>1</staff>
</note>
<note color="#000000" default-x="141">
<pitch>
<step>B</step>
<octave>4</octave>
</pitch>
<duration>256</duration>
<instrument id="P1-I1" />
<voice>1</voice>
<type>quarter</type>
<stem>down</stem>
<staff>1</staff>
</note>
<note color="#000000" default-x="205">
<pitch>
<step>A</step>
<octave>4</octave>
</pitch>
<duration>256</duration>
<instrument id="P1-I1" />
<voice>1</voice>
<type>quarter</type>
<stem>up</stem>
<staff>1</staff>
</note>
</measure>
<measure number="4" width="178">
<note color="#000000" default-x="15">
<pitch>
<step>G</step>
<octave>4</octave>
</pitch>
<duration>1024</duration>
<instrument id="P1-I1" />
<voice>1</voice>
<type>whole</type>
<stem>up</stem>
<staff>1</staff>
</note>

```

Beberapa bagian yang akan kita gunakan pada pengecekan ini adalah sebagai berikut :

- Pitch : terdiri dari step dan octave. Bagian ini merepresentasikan bunyi dari suatu not
- Divisions : melambangkan panjang satu ketukan
- Duration : melambangkan panjang satu not dibunyikan
- Properti default-x dari note : merepresentasikan

posisi not tersebut.

- Alter : tidak terdapat pada contoh ini, namun merepresentasikan tnda kres / mol pada musik.

IV. PROSES PEMBANDINGAN

Sebagai contoh dari perbandingan yang akan terjadi, penulis akan membandingkan not balok yang ada pada Gambar 5 dengan not balok pada Gambar 6 berikut ini :



Gambar 6

Gambar 6 adalah suatu bagian musik yang ingin kita periksa kemiripannya dengan musik pada Gambar 5. Ada beberapa langkah yang harus kita ambil untuk melakukan perbandingan, yaitu :

- Tokenisasi tahap-1 : Mengubah bentuk musicXML dari masing-masing musik untuk menjadi bagian yang lebih mudah dibandingkan
- Tokenisasi tahap-2 : Mengubah bentuk yang telah ditokenisasi tadi menjadi interval-interval perubahan nada pada setiap ketukan.
- Perbandingan dengan menggunakan Algoritma Boyer-Moore

Tahap pertama yang harus ditempuh adalah tokenisasi. Tidak semua informasi dari musicXML kita butuhkan untuk mendeteksi plagiarisme musik. Maka, bagian yang kita ambil hanyalah pitch, division, duration, alter, dan default-x. Bagian-bagian ini masih dapat kita simplifikasi lagi menjadi sebuah tipe bentukan yang terdiri dari :

- ArrayList / Vector of Notes : Notes yang dimaksud di sini adalah step dan octave yang dapat direpresentasikan dengan tipe data String. Contoh isi dari data ini adalah A4\,(C4,C4/). Garis '/' atau '\' setelah step yang ditulis adalah perlambangan untuk kres atau mol, di mana '\' melambangkan mol sedangkan '/' melambangkan kres.
- Duration : Diambil dari type, melambangkan berapa ketuk not tersebut dimainkan. Hasil yang disimpan dapat direpresentasikan dengan tipe data Float, dimana nilai yang disimpan adalah sejumlah duration sebuah note dibagi dengan divisions.

Sebagai contoh, ketika musicXML dari Gambar 5 selesai ditokenisasi, hasilnya adalah sebagai demikian :

G5	E5	F5	G5	C5	D5	C5	B4	A4	G4
2	1	1	2	2	1	1	1	1	4

Table 1 Hasil Tokenisasi Gambar 5

Sedangkan musicXML dari Gambar 6, setelah selesai ditokenisasi, hasilnya adalah sebagai demikian :

A5	D5	E5	D5	C5/	B4
2	2	1	1	1	1

Table 2 Hasil Tokenisasi Gambar 6

Sampai di sini, tokenisasi tahap-1 telah selesai.

Pada tokenisasi tahap-2, penulis akan mengubah hasil

tokenisasi tahap-1 menjadi array of float sederhana. Perubahan yang dilakukan adalah :

- Mengubah susunan not menjadi susunan perubahan interval. Sebagai contoh, antara nada G dan E pada table 1 memiliki interval -1,5.
- Membuat seluruh susunan tadi memiliki panjang interval yang sama. Sebagai contoh, nada A pada Table 2 dapat diubah dari {A,2} menjadi {A,1} dan {A,1}. Panjang interval yang dijadikan acuan adalah panjang interval yang terpendek dari kedua table.

Sebagai dasar teori perubahan interval, perlu diketahui bahwa secara umum suatu musik terdiri dari 12 jenis note : C, C/, D, D/, E, F, F/, G, G/, A, A/, B. Untuk setiap step yang penulis tuliskan sebelumnya, untuk octave yang sama, terdapat interval 0.5 antar step. Untuk suatu oktaf yang berbeda, terdapat perbedaan interval sebanyak 6. Sebagai contoh, C4 dan C5 memiliki perbedaan interval 6. Namun C5 dan B4 memiliki perbedaan interval -0.5.

Setelah tokenisasi tahap-2, hasilnya untuk table 1 adalah sebagai berikut :

0	0	-1.5	0.5	1	0	-3.5	0	1	-1	-0.5	-1	-1	0	0	0
---	---	------	-----	---	---	------	---	---	----	------	----	----	---	---	---

Cara membaca table di atas adalah sebagai berikut :

- Pada ketukan ke-0, terdapat interval 0 perubahan nada
- Pada ketukan ke-1, terdapat interval 0 perubahan nada
- Pada ketukan ke-2, terdapat interval -1.5 perubahan nada.

Sedangkan hasil tokenisasi tahap-2 untuk table 2 adalah sebagai berikut :

0	0	-3.5	0	1	-1	-0.5	-1
---	---	------	---	---	----	------	----

Untuk setiap table hasil tokenisasi, seluruh nilai 0 pertama dapat diabaikan hingga nilai yang tidak bernilai 0. Hal ini dikarenakan nada pertama tidak akan memiliki perubahan interval dari nada sebelumnya, mengingat tidak ada nada sebelum nada pertama. Angka 0 dibelakannya dibuang karena selama interval bernilai 0, maka nada yang dibunyikan masih merupakan nada pertama yang sama bunyinya. Maka array of integer yang dibandingkan menjadi :

-1.5	0.5	1	0	-3.5	0	1	-1	-0.5	-1	-1	0	0	0
------	-----	---	---	------	---	---	----	------	----	----	---	---	---

dan

-3.5	0	1	-1	-0.5	-1
------	---	---	----	------	----

Langkah tokenisasi tahap-2 selesai di sini. Setelah tokenisasi tahap-2 ini selesai, maka kita dapat menggunakan algoritma Boyer-Moore seperti biasa. Jika pattern dari table 2 ditemukan pada table 1, maka dapat disimpulkan terdapat bagian dari musik table-2 yang merupakan bagian dari musik table-1. Hasil ini dapat menjadi parameter apakah suatu lagu merupakan plagiasi atau bukan. Hasil dari contoh perbandingan ini akan menemukan pattern table-2 pada elemen ke 4 table-1,

sehingga dapat disimpulkan bahwa ada bagian dari musik table-2 yang sangat mirip dengan musik pada table-1.

Langkah-langkah yang telah dilakukan penulis tadi adalah penanganan untuk kasus plagiasi musik-musik yang hanya memainkan nada-nada tunggal / singular. Namun bagaimana dengan musik-musik lain seperti musik orkestra, ataupun musik band?

Untuk kasus-kasus yang telah disebutkan di atas, ada beberapa solusi yang dapat kita ambil, antara lain :

- Membuat tokenisasi untuk part alat musik yang sesuai saja
- Membuat tokenisasi untuk seluruh part alat musik, kemudian mencocokkan pattern yang dicari ke seluruh hasil tokenisasi
- Membuat analisa berdasarkan harmoni yang ditimbulkan oleh kombinasi-kombinasi nada dari beberapa alat musik yang dibunyikan secara bersamaan.

Di antara ketiga solusi di atas, tidak ada perbedaan signifikan antara cara 1 dan 2 dengan cara yang penulis gunakan sebelumnya untuk mendeteksi plagiarisme. Namun untuk cara 3, akan perlu dijelaskan lebih lanjut.

Harmoni-harmoni yang dibentuk oleh suatu alat musik / kumpulan alat musik biasanya dapat kita sebut chord. Chord minimal terdiri dari 3 nada. Contoh-contoh chord yang standar antara lain adalah sebagai berikut :

- C : C E G
- D7 : D F/ A C
- Am7 : A C E G

Oleh karena model analisis yang berbeda, maka ada tahap pula yang berubah dalam melakukan tokenisasi. Tokenisasi tahap-1 tetap berjalan dengan cara yang sama, namun terdapat perubahan pada tokenisasi tahap-2.

Sebagai contoh, misalkan saja hasil tokenisasi tahap 1 adalah sebagai berikut :

C	F	C	A
4	4	4	4

E	A	G	C
4	4	4	4

G	C	E	E
4	4	4	4

Maka untuk tokenisasi tahap-2, setiap nada yang berada pada ketukan yang sama akan digabungkan dan dianalisis chordnya. Berdasarkan contoh yang ada di atas, kumpulan nada yang dianalisis adalah : {C,E,G}, {F,A,C}, {C,G,E}, {A,C,E}. Setelah dicocokkan ke database chord, akan ditemukan bahwa chord-chord yang dimainkan pada contoh di atas adalah C, F, C, Am.

Tokenisasi tahap-2 yang dilakukan akan memiliki sedikit perbedaan :

- Nilai yang dicatat bukan perubahan notnya, namun perubahan chord beserta intervalnya.
- Perubahan chord yang dicatat adalah jenis

chordnya. Salah satu cara merepresentasikannya :

- Major -> M
- Minor -> m
- Diminished -> d
- Dominant 7th -> 7
- Selain itu, sisanya sama dengan sebelumnya

Maka, hasil tokenisasi tahap-2 dari contoh di atas adalah sebagai berikut :

0	-3.5M	3.5 M	-1.5m
---	-------	-------	-------

Selanjutnya perbandingan dapat dilakukan seperti biasa cara pertama dengan menggunakan Algoritma Boyer-Moore.

V. KEMUNGKINAN PENGEMBANGAN

Pada karya tulis ini, penulis hanya membahas aplikasi penggunaan Boyer-Moore untuk musik yang tertulis. Namun, sebenarnya solusi yang telah dibahas oleh penulis dapat juga digunakan untuk mendeteksi plagiarisme musik live yang tidak tertulis, dengan perbedaan pada cara tokenisasi tahap-1nya, karena perbedaan tipe file yang dibuka.

Sebagai contoh, jika kita ingin mendeteksi plagiarisme musik live, kita dapat menggunakan Algoritma Fast-Fourier Transform terlebih dahulu untuk mengetahui frekuensi nada yang dimainkan, sebelum kemudian bisa dilanjutkan ke tahap tokenisasi berikutnya. Contoh lain adalah ketika ingin mendeteksi plagiarisme pada musik .midi, kita bisa membuka file midi yang bersangkutan dan kemudian melakukan tokenisasi pada file tersebut.

Selain itu, bila kita lihat, sebenarnya kombinasi karakter yang ada tidak terlalu banyak. Maka dari itu, penggunaan algoritma brute force mungkin akan lebih menguntungkan ketimbang menggunakan algoritma Boyer-Moore. Hal ini disebabkan karena untuk waktu yang tidak berbeda jauh, algoritma Brute Force dapat dimodifikasi untuk menghitung derajat kesamaan; Sesuatu yang tidak dapat dilakukan dengan Algoritma Boyer Moore.

VI. KESIMPULAN

Kesimpulan yang dapat diambil penulis adalah sebagai berikut :

- Algoritma Boyer-Moore bisa digunakan tidak terbatas hanya pada pattern-matching string saja seperti yang diketahui selama ini.
- Plagiarisme musik dapat dideteksi dengan lebih baik dengan bantuan Algoritma Boyer-Moore.
- Algoritma Boyer-Moore bukan satu-satunya metoda pattern matching yang dapat digunakan dalam pendeteksian plagiarisme musik.

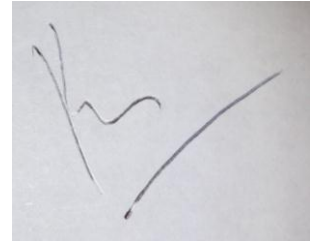
REFERENCES

- [1] <http://www-igm.univ-mlv.fr/~lecroq/string/node14.html>
- [2] R. Munir, *Diktat Kuliah IF 3051 Strategi Algoritma*. Bandung: Penerbit Sekolah Teknik Elektro dan Informatika, 2009.

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 Desember 2012



Nicholas Rio (13510024)