

PENCARIAN MELODI PADA FILE MIDI

Eddo Fajar N

Peneliti di Puslitbang Aptel SKDI, Badan Litbang SDM Kementerian Kominfo, Alumnus
Pasca Sarjana Fakultas Teknik Elektro ITB Bandung

ABSTRACT

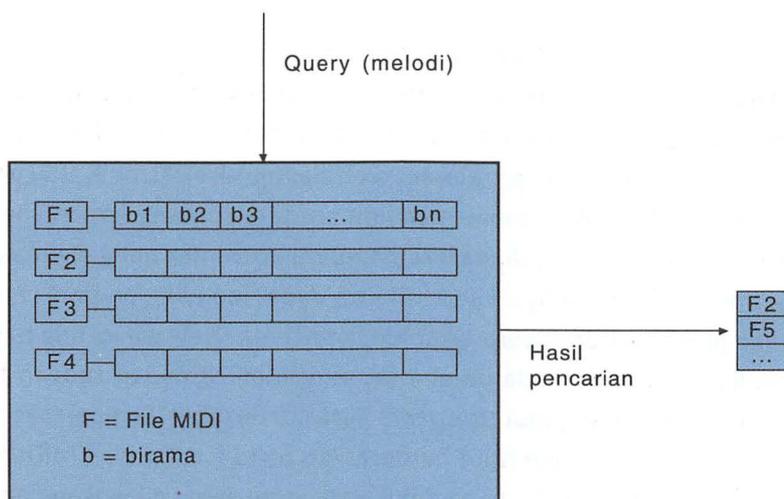
Searching MIDI files is generally done by looking at their file names or the metadata/tags inside them. Searching for a certain melody in a MIDI file must be done by looking at the contents on the MIDI file. When searching for a melody in a MIDI file, the user inputs a melody as a query. The system will then search for MIDI files that contain the most similar melodies compared to the query. The queried melody is monophonic while melodies contained in MIDI files are polyphonic. Therefore, melodies in the MIDI files must be converted to monophonic in order to enable comparison. The comparison of two monophonic melodies is done by calculating their dissimilarity factor. The smaller the dissimilarity factor, the more similar the two melodies are. A zero dissimilarity factor means the two melodies are identical. There are two comparison methods used, i.e. bar per bar comparison and note per note comparison. Bar per bar comparison is faster but is very fussy about the position of the melody in the bar. Note per note comparison is more accurate but takes longer to search.

PENDAHULUAN

MIDI (*Musical Instrument Digital Interface*) adalah protokol yang memungkinkan alat musik elektronik dan perangkat lunak komputer untuk berkomunikasi. Isi file MIDI dibuat segenerik mungkin supaya dapat diakses oleh alat musik elektronik atau perangkat lunak musik manapun. File MIDI dapat menyimpan data tentang kapan suatu nada dibunyikan atau dimatikan, tempo, tanda birama, suara instrumen yang digunakan pada masing-masing *track* dan sebagainya. File MIDI juga dapat menyimpan data yang hanya perlu diakses oleh alat musik atau perangkat lunak tertentu. Dewasa ini ada banyak file MIDI yang tersedia di internet. Pada umumnya, pencarian terhadap sebuah file MIDI dilakukan berdasarkan beberapa judul atau *genre* file MIDI tersebut. Sampai saat ini belum ada pencarian terhadap sebuah file MIDI berdasarkan melodi dalam file MIDI tersebut. Pencarian terhadap file MIDI berdasarkan judul atau *genre* dapat dilakukan dengan cara melihat nama file atau melihat *header* di dalam file MIDI tersebut. Pencarian terhadap file MIDI berdasarkan melodi yang diinginkan harus dilakukan berdasarkan isi (*content*) dari file MIDI tersebut. Pencarian terhadap file MIDI berdasarkan melodi memiliki beberapa kegunaan, misalnya: (1) Untuk mengetahui judul lagu yang memiliki melodi yang dimaksud.

(2) Untuk mencari apakah sudah ada lagu yang memiliki melodi yang mirip (atau sama) dengan melodi yang dimaksud. Ini biasanya digunakan oleh para pencipta lagu untuk mengetahui apakah ada lagu lain yang memiliki melodi yang sama dengan melodi dari suatu lagu tertentu. Jika ternyata ditemukan lagu lain yang memiliki melodi yang mirip sepanjang 8 birama atau lebih, maka dapat disimpulkan bahwa lagu tersebut adalah lagu jiplakan.

Pada pencarian terhadap file MIDI berdasarkan melodi, pengguna memasukkan *query* berupa melodi dan sistem akan mencari file MIDI mana saja yang memiliki potongan melodi yang paling mirip dengan melodi yang di-*query* kan oleh pengguna. Ada beberapa penelitian mengenai *music information retrieval* yang sudah pernah dilakukan sebelumnya. Yongwei Zhu, Mohan Kankanhalli dan Qi Tian dari National University Of Singapore melakukan penelitian mengenai pencarian lagu dengan cara menggumamkan (*humming*) melodi lagu tersebut [1]. Perbandingan melodi dilakukan dengan cara menganalisis kontur melodi. Hung-Ming Yu, Wei-Ho Tsai dan Hsin-Min Wang dari Academia Sinica, Taiwan, melakukan penelitian mengenai pencarian lagu pada sistem karaoke berdasarkan *query-by-singing*. Melodi yang dijadikan *query* dimasukkan dengan cara menyanyikan melodi tersebut [2]. File MIDI dapat dianalogikan dengan dokumen dalam sebuah sistem *information retrieval* biasa. Data pada tiap birama untuk masing-masing *channel* pada file MIDI dapat dianalogikan dengan *term*. Skema pencarian melodi dapat dilihat pada ilustrasi pada Gambar 1 berikut.



Gambar 1. Skema Dasar Pencarian Melodi

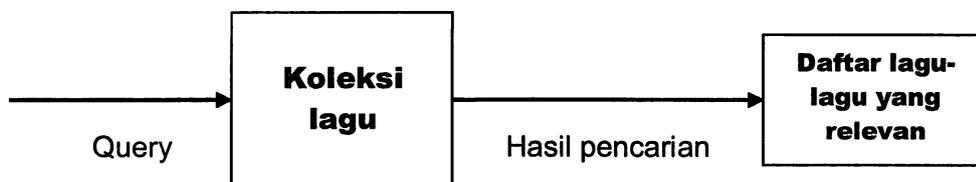
Query yang dimasukkan bisa terdiri dari hanya satu birama atau bisa lebih dari satu birama. Sistem akan mencari birama (atau birama-birama yang berurutan) mana saja yang berisi melodi yang paling mirip dengan melodi yang dijadikan *query*. File MIDI hasil pencarian tidak harus memiliki melodi yang sama persis dengan melodi yang dijadikan *query*, tapi harus memiliki melodi yang mirip dalam batas toleransi tertentu.

Untuk tiap *query* yang diajukan oleh pengguna, sistem akan melakukan penghitungan derajat kemiripan terhadap tiap birama pada tiap file MIDI di dalam sistem berdasarkan *query* tersebut. Jika nilai derajat kemiripan yang dihasilkan semakin kecil, maka melodi yang terkandung dalam melodi birama semakin mirip dengan melodi dalam *query*. Jika nilai derajat kemiripan sama dengan 0, maka melodi yang terkandung dalam birama sama dengan melodi dalam *query*. Melodi yang dijadikan *query* bersifat monofonik sedangkan melodi pada file MIDI bersifat polifonik. Karena itu, melodi pada file MIDI perlu dikonversi menjadi melodi monofonik supaya dapat dibandingkan dengan melodi yang dijadikan *query*. Alexandra L. Uitdenbogerd dan Justin Zobel dari Royal Melbourne Institute Of Technology, Australia, melakukan penelitian mengenai algoritma ekstraksi melodi polifonik dari file MIDI menjadi melodi monofonik [3].

PEMBAHASAN

Music Information Retrieval

Music Information retrieval (MIR) adalah disiplin ilmu yang merupakan gabungan antara *information retrieval* (IR) dan musik. Tujuan MIR adalah untuk mencari lagu yang relevan dengan sebuah *query* tertentu. Gambaran umum sebuah sistem MIR dapat dilihat pada Gambar 2.



Gambar 2. Gambaran Umum Sistem *Music Information Retrieval*

Pada gambar di atas, *query* dapat berupa melodi, genre lagu, lirik, tempo, irama dan sebagainya. Lagu-lagu dalam koleksi lagu dapat disimpan dalam berbagai macam format file audio, misalnya wav, mp3, MIDI, *real audio* dan sebagainya. *Music information*

retrieval adalah disiplin ilmu yang relatif baru. Pada awalnya orang melakukan pencarian terhadap sebuah lagu berdasarkan *tag* metadata berbasis teks yang mengidentifikasi fitur-fitur tertentu, misalnya judul atau pencipta lagu. Pencarian seperti ini menyulitkan bagi orang yang tidak tahu isi dari *metadata* yang akan dicari. Ada kalanya seseorang tidak mengetahui *metadata* dari sebuah lagu tapi mengetahui fitur-fitur lainnya seperti melodi, instrumentasi atau harmonisasi dari lagu tersebut. Akan lebih baik jika pencarian pada terhadap sebuah lagu dilakukan berdasarkan isi (*content based*) dari lagu tersebut. Inilah yang melatarbelakangi lahirnya *music information retrieval*.

Pencarian terhadap sebuah lagu secara *content based* pertama kali dilakukan berdasarkan melodi dari lagu tersebut. Aplikasi Themefinder [8], yang disponsori oleh Stanford University, menyimpan sekitar 20,000 lagu klasik, barok, renaissance dan lagu-lagu rakyat (*folk song*) Eropa. Pengguna melakukan pencarian terhadap sebuah lagu dengan cara memasukkan informasi mengenai kontur melodi pada sebuah antarmuka berbasis teks. Aplikasi Meldex [9], yang merupakan projek dari University Of Waikato, Selandia Baru, menyimpan sekitar 9,000 lagu rakyat Cina dan Eropa. Pengguna melakukan pencarian dengan cara memainkan melodi masukan pada sebuah kibord maya di halaman *web* Meldex. Meldex akan mengembalikan lagu-lagu yang dianggap memiliki melodi yang mirip dengan melodi masukan yang terurut berdasarkan kemiripan kepada pengguna. Aplikasi Shazam [11] melakukan pencarian dengan cara *query by example*. Pengguna memasukkan sampel lagu selama 10 detik. Shazam mengembalikan lagu-lagu yang mirip dengan sampel tersebut. Sampai saat ini pencarian terhadap sebuah lagu yang bersifat *content based* baru dilakukan berdasarkan melodi. Pada masa yang akan datang, diramalkan bahwa pencarian terhadap sebuah lagu dapat dilakukan berdasarkan aspek-aspek lain seperti instrumentasi, harmoni, lirik, tanda birama, genre dan sebagainya.

File MIDI

MIDI adalah singkatan dari *Musical Instrument Digital Interface* dan sangat populer di kalangan musisi sejak ditemukan pertama kali pada tahun 1983. MIDI merupakan suatu protokol komunikasi yang memungkinkan alat-alat musik elektronik berkomunikasi satu sama lain. MIDI merupakan salah satu protokol yang paling efisien untuk merepresentasikan informasi musik. Hal ini menyebabkan protokol MIDI tidak hanya bermanfaat bagi para musisi, melainkan juga untuk aplikasi komputer yang menghasilkan suara seperti *multimedia* atau *game*. SMF (*Standard MIDI File*) adalah format file yang dirancang khusus untuk menyimpan data yang dimainkan atau direkam oleh sebuah alat musik, baik yang berbasis perangkat keras maupun yang berbasis perangkat lunak. Sebuah file MIDI menyimpan *event* yang mengatur nada-nada yang dimainkan, pemilihan alat musik yang dimainkan, tempo, tanda birama, tanda kunci dan sebagainya. Sebuah file

MIDI menyimpan *event* yang menyatakan nada-nada yang dimainkan dalam beberapa *channel*.

Sebuah file MIDI dapat menampung hingga 16 *channel*. Selain itu, file MIDI juga dapat menyimpan informasi-informasi musik lainnya (termasuk nada-nada yang dimainkan pada sebuah *channel* tertentu) dalam beberapa *track*. Sebuah file MIDI dapat menampung hingga 65025 *track*. Sebuah *track* dapat dianalogikan dengan satu bagian dalam musik, seperti bagian trompet. Sebuah *pattern* dapat dianalogikan dengan semua bagian musik (trompet, piano, drum, dan sebagainya) pada sebuah lagu. Format SMF dirancang segenerik mungkin agar alat musik elektronik manapun dapat membaca atau membuat file dengan format SMF tanpa harus kehilangan data-data yang terpenting. Format SMF juga dirancang sefleksibel mungkin agar sebuah alat musik dapat menyimpan data yang hanya dapat dimengerti oleh alat musik itu sendiri dan tidak perlu dibaca oleh alat musik lain.

Konversi Melodi Polifonik Menjadi Melodi Monofonik

Ada beberapa algoritma yang dapat digunakan untuk mengonversi melodi polifonik menjadi melodi monofonik, yaitu [3]:

1. Pada alternatif algoritma pertama, semua nada yang dibunyikan pada file MIDI digabungkan ke dalam satu kumpulan *event*. Jika ada beberapa nada yang dibunyikan pada saat bersamaan, maka nada yang dipilih adalah nada yang paling tinggi. Pada algoritma ini, panjang nada tidak diperhatikan. Jika kemudian ada kumpulan nada lain yang dibunyikan, maka nada yang dipilih adalah nada tertinggi dari kumpulan nada yang baru meskipun mungkin nada terpilih yang lama lebih tinggi daripada kumpulan nada yang baru. Sebagai contoh, melodi pada Gambar 4 adalah hasil penerapan algoritma ini terhadap melodi pada Gambar 3. Seperti terlihat pada Gambar 4, kelemahan algoritma ini adalah kemungkinan munculnya nada-nada tambahan yang tidak diinginkan.



Gambar 3. Contoh Melodi



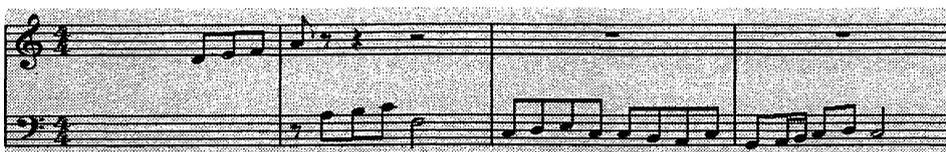
Gambar 4. Hasil Penerapan Alternatif Algoritma 1

2. Pada alternatif algoritma kedua, setiap *channel* pada file MIDI diproses secara terpisah. Melodi pada masing-masing *channel* diproses seperti pada algoritma pertama. Melodi yang dipilih adalah melodi pada *channel* yang memiliki rata-rata nada tertinggi. Kelemahan algoritma ini adalah mungkin saja melodi yang dipilih adalah melodi iringan (*accompaniment*) yang kebetulan bernada tinggi. Hasil penerapan algoritma ini terhadap melodi pada Gambar 3 dapat dilihat pada Gambar 5.



Gambar 5. Hasil Penerapan Alternatif Algoritma II

3. Pada alternatif algoritma ketiga, setiap *channel* pada file MIDI diproses secara terpisah. Melodi pada masing-masing *channel* diproses seperti pada algoritma pertama. Jika ada beberapa nada yang dibunyikan pada saat bersamaan, maka nada yang dipilih adalah nada yang paling rendah. Kelemahan algoritma ini adalah mungkin saja melodi yang dipilih adalah melodi iringan jika pada sebuah *channel* terdapat melodi utama dan melodi iringan. Hasil penerapan algoritma ini terhadap melodi pada Gambar 3 dapat dilihat pada Gambar 6.



Gambar 6. Hasil Penerapan Alternatif Algoritma III

4. Pada alternatif algoritma keempat, setiap *channel* pada file MIDI diproses secara terpisah. Melodi pada masing-masing *channel* diproses seperti pada algoritma pertama. Melodi yang dipilih adalah *channel* yang memiliki entropi melodi tertinggi. Hasil penerapan algoritma ini terhadap melodi pada Gambar 3 dapat dilihat pada Gambar 7.



Gambar 7. Hasil Penerapan Alternatif Algoritma IV

Pencarian Melodi

Metode-metode yang digunakan untuk pencarian melodi dapat dibagi dalam tiga kategori, yaitu:

1. Pencarian melodi berbasis *index term*. Pengindeksan dapat meningkatkan performansi sebuah sistem IR karena semua informasi yang relevan sudah dibangkitkan secara *offline*. Operasi pencarian dilakukan dengan cara membandingkan *query* dengan indeks-indeks dalam sebuah dokumen. Pencarian dengan metode ini juga dapat diterapkan pada sebuah sistem MIR. Indeks dalam hal ini dapat dianalogikan dengan potongan melodi yang diekstraksi dari sebuah lagu. Salah satu metode ekstraksi melodi yang paling banyak digunakan adalah metode *N-gram*, yaitu dengan mengelompokkan setiap *N* nada yang berurutan menjadi satu segmen melodi. S. Downie dan M. Nelson [12], mencoba melakukan eksperimen dengan mengekstraksi lagu-lagu rakyat menjadi potongan-potongan melodi dengan metode *N-gram*. Eksperimen ini menghasilkan hasil pencarian yang bagus untuk *query* yang panjang. S. Doraisamy dan G. Widmer [13] mencoba melakukan pencarian melodi terhadap lagu-lagu polifonik dengan metode *N-gram* tanpa melakukan konversi ke melodi monofonik

2. Pencarian melodi berbasis *sequence matching*. Pada pencarian melodi berbasis *sequence matching*, melodi direpresentasikan dengan deretan angka-angka. Pencarian dilakukan dengan membandingkan *query* dengan deretan angka-angka tersebut. Salah satu metode pencarian melodi berbasis *sequence matching* adalah dengan memperhatikan kontur melodi. Perpindahan dari satu nada ke nada lain dapat dibagi menjadi tiga jenis, yaitu: menaik, menurun dan tetap. A. Ghias, J. Logan, D. Chamberlain dan B.C. Smith [14] menggunakan metode ini untuk menangani ketidakcocokan yang mungkin timbul antara *query* yang dimasukkan dengan cara *humming* dengan lagu-lagu yang dibandingkan.

Metode lain yang berbasis *sequence matching* adalah dengan cara menghitung *edit distance*. *Edit distance* antara dua melodi dapat didefinisikan sebagai jumlah minimum biaya (*cost*) yang diperlukan untuk mentransformasikan melodi sumber menjadi melodi sasaran dengan operasi-operasi penyuntingan yang tersedia. Operasi-operasi penyuntingan pada umumnya meliputi:

- Penyisipan, yaitu menyisipkan sebuah nada ke melodi sasaran.
- Penghapusan, yaitu menghapus sebuah nada dari melodi sumber.
- Penggantian, yaitu mengganti sebuah nada pada melodi sumber dengan nada dari melodi sasaran.

Pada umumnya, pencarian melodi berbasis *sequence matching* hanya memperhatikan *pitch* dan durasi nada. M. Robine, P. Hanna dan P. Ferraro [15] mencoba memperbaiki hasil pencarian dengan menghitung *edit distance* antara dua buah melodi dengan cara memperhitungkan elemen-elemen musik lainnya seperti tonalitas dan irama.

3. Pencarian melodi berbasis *geometry matching*. Pada pencarian melodi berbasis *geometry matching*, perbandingan melodi dilakukan dengan cara mentranslasikan *query* dengan melodi yang direpresentasikan secara geometrik. Representasi melodi secara geometrik yang paling sering digunakan orang adalah representasi *piano roll*. Pada representasi *piano roll*, melodi dipetakan ke dalam diagram fungsi *pitch* terhadap waktu.

Sistem Pencarian Melodi pada File MIDI

Secara garis besar, sistem akan menerima masukan *query* berupa melodi monofonik, kemudian sistem akan mencari file-file MIDI mana saja dari sebuah kumpulan file MIDI yang memiliki melodi yang mirip dengan melodi yang dijadikan *query*. *Query* yang dimasukkan bisa terdiri dari hanya satu birama atau bisa lebih dari satu birama. Sistem akan mencari birama (atau birama-birama yang berurutan) mana saja yang berisi melodi yang paling mirip dengan melodi yang dijadikan *query*. File MIDI hasil pencarian tidak harus memiliki melodi yang sama persis dengan melodi yang dijadikan *query*, tapi harus memiliki melodi yang mirip dalam batas toleransi tertentu.

Untuk tiap *query* yang diajukan oleh pengguna, sistem akan melakukan penghitungan derajat ketidakmiripan terhadap tiap birama pada tiap file MIDI di dalam sistem berdasarkan *query* tersebut. Jika nilai derajat ketidakmiripan yang dihasilkan semakin kecil, maka melodi yang terkandung dalam melodi birama semakin mirip dengan melodi dalam *query*. Jika nilai derajat ketidakmiripan sama dengan 0, maka melodi yang terkandung dalam birama sama dengan melodi dalam *query*.

Penghitungan derajat ketidakmiripan untuk setiap file MIDI ditempuh dalam tiga proses, yaitu:

1. Konversi melodi pada file MIDI menjadi melodi monofonik.
2. Pemetaan melodi hasil konversi dari file MIDI dan melodi yang dijadikan *query* ke diagram fungsi interval terhadap waktu.
3. Perbandingan melodi pada file MIDI dengan melodi pada *query*.

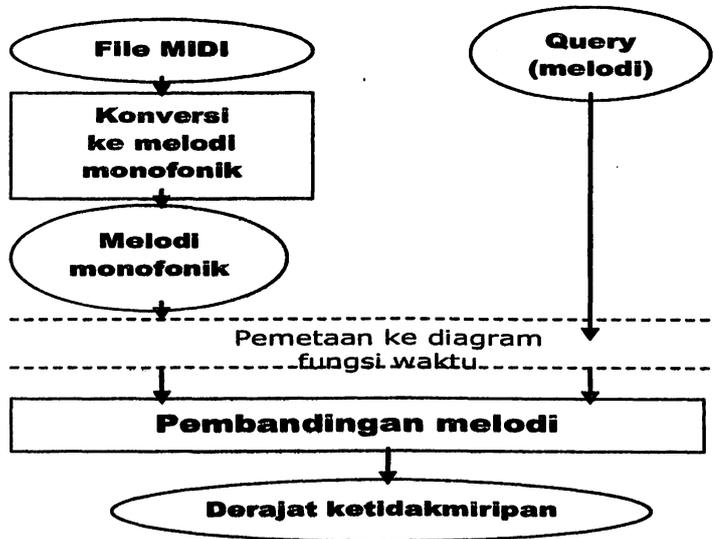
Berdasarkan segmen melodi yang dibandingkan, ada dua cara untuk membandingkan melodi pada file MIDI dengan melodi pada *query*, yaitu:

1. Perbandingan melodi per birama. Sistem akan membandingkan melodi pada tiap birama dalam tiap file MIDI dengan melodi yang dijadikan *query*. Kelebihan perbandingan melodi per birama adalah waktu eksekusinya yang relatif cepat karena suatu lagu biasanya hanya terdiri dari puluhan birama. Kekurangannya adalah nada pertama *query* harus berada di ketukan pertama pada birama.
2. Perbandingan melodi per nada. Sistem akan membandingkan melodi dengan tiap nada dalam tiap file MIDI sebagai nada awal melodi dengan melodi yang dijadikan *query*. Panjang potongan melodi dari file MIDI yang dibandingkan sama dengan

panjang *query*. Kelebihan pembandingan melodi per nada adalah fleksibilitas; nada pertama *query* tidak perlu berada di ketukan pertama pada birama. Kekurangannya adalah waktu eksekusinya yang relatif lebih lama karena suatu lagu biasanya terdiri dari ratusan nada.

Skema penghitungan derajat ketidakmiripan dapat dilihat pada Gambar 8. Langkah-langkah yang ditempuh dalam proses penghitungan derajat ketidakmiripan untuk setiap file MIDI adalah sebagai berikut:

1. Sistem mengonversi melodi dari file MIDI yang sedang diproses menjadi melodi monofonik. Melodi hasil konversi inilah yang akan dibandingkan dengan melodi yang dijadikan *query*.
2. Sistem memetakan melodi yang dijadikan *query* ke diagram fungsi interval terhadap waktu.
3. Sistem memetakan melodi hasil konversi dari file MIDI yang sedang dibuka ke diagram fungsi interval terhadap waktu.
4. Pada pembandingan melodi yang dilakukan per birama, nyatakan panjang melodi yang dijadikan *query* sebagai b_1 (dalam satuan birama) dan panjang file MIDI sebagai b_2 . Sistem akan melakukan pembandingan terhadap setiap b_1 birama yang berurutan pada melodi pada file MIDI dengan melodi yang dijadikan *query* berdasarkan diagram fungsi interval terhadap waktu yang telah dihasilkan sebelumnya. Pembandingan ini akan menghasilkan nilai derajat ketidakmiripan antara kedua melodi tersebut. Pembandingan melodi dilakukan dari birama pertama sampai birama ke- $(b_2 - b_1 + 1)$ pada file MIDI. Pada pembandingan melodi yang dilakukan per nada, nyatakan panjang melodi yang dijadikan *query* sebagai b_1 (dalam satuan birama). Sistem akan melakukan pembandingan terhadap setiap melodi yang diawali oleh tiap nada dalam file MIDI dan memiliki panjang b_1 birama dengan melodi yang dijadikan *query* berdasarkan diagram fungsi interval terhadap waktu yang telah dihasilkan sebelumnya. Pembandingan ini akan menghasilkan nilai derajat ketidakmiripan antara kedua melodi tersebut. Pembandingan melodi dilakukan dari nada pertama sampai nada terakhir pada file MIDI.
5. Hasil dari pencarian melodi untuk file MIDI yang sedang diproses adalah melodi pada b_1 birama yang berurutan pada file MIDI yang menghasilkan nilai derajat ketidakmiripan terkecil.



Gambar 8. Skema Penghitungan Derajat Ketidakmiripan

Secara garis besar, *pseudocode* algoritma penghitungan derajat ketidakmiripan pada sebuah file MIDI dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1. Algoritma Penghitungan Derajat Ketidakmiripan Per Birama

```

Ubah melodi pada file MIDI menjadi melodi monofonik  $m$ 
i traversal [1 .. ( $b_2 - b_1 + 1$ )]
DerajatKetidakmiripan  $\leftarrow$  hasil perbandingan melodi antara melodi
query dengan melodi  $m$  dari birama  $i$  sampai ( $i + b_1 - 1$ )
If  $i = 1$  then
  DerajatKetidakmiripanMinimum  $\leftarrow$  DerajatKetidakmiripan
Else {  $i > 1$  }
  If DerajatKetidakmiripan < DerajatKetidakmiripanMinimum then
    DerajatKetidakmiripanMinimum  $\leftarrow$  DerajatKetidakmiripan
  {Next  $i$ }
Return DerajatKetidakmiripanMinimum
  
```

Tabel 2. Algoritma Penghitungan Derajat Ketidakmiripan Per Nada

```
Ubah melodi pada file MIDI menjadi melodi monofonik  $m$ 
i traversal [1 .. jumlah nada dalam  $m$ ]
DerajatKetidakmiripan hasil pembandingan melodi antara melodi
query dengan potongan melodi  $m$  yang diawali oleh nada ke- $i$ 
dari  $m$  dan memiliki panjang  $b1$  birama
If  $i = 1$  then
    DerajatKetidakmiripanMinimum  $\leftarrow$  DerajatKetidakmiripan
Else  $\{i > 1\}$ 
    If DerajatKetidakmiripan  $<$  DerajatKetidakmiripanMinimum then
        DerajatKetidakmiripanMinimum  $\leftarrow$  DerajatKetidakmiripan
    {Next  $i$ }
Return DerajatKetidakmiripanMinimum
```

Deskripsi untuk masing-masing proses pada Gambar 8 akan dijelaskan di bawah ini.

Konversi Melodi Pada File MIDI Menjadi Melodi Monofonik

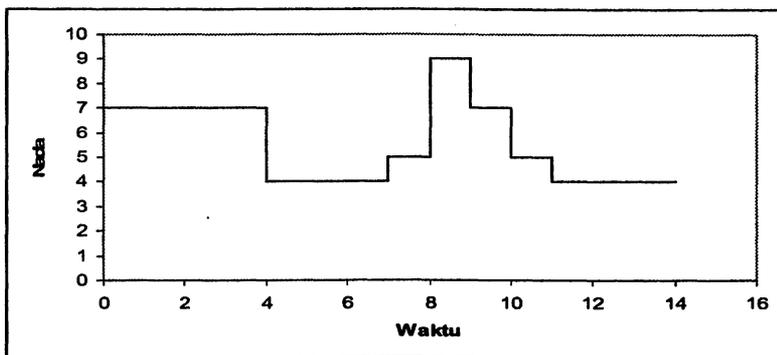
Melodi yang dijadikan *query* bersifat monofonik sedangkan melodi pada file MIDI bersifat polifonik. Karena itu, melodi pada file MIDI perlu dikonversi menjadi melodi monofonik supaya dapat dibandingkan dengan melodi yang dijadikan *query*. Algoritma-algoritma konversi melodi polifonik ke monofonik yang digunakan adalah algoritma-algoritma yang telah disebutkan sebelumnya.

Pemetaan Melodi Ke Diagram Fungsi Interval Terhadap Waktu

Pada proses ini akan dilakukan pemetaan terhadap melodi yang dijadikan *query* dan melodi dari file MIDI yang telah dikonversi menjadi melodi monofonik ke dalam diagram fungsi interval terhadap waktu. Melodi dapat didefinisikan sebagai susunan nada-nada yang terurut berdasarkan waktu kemunculannya. Berdasarkan definisi di atas, melodi dapat dimodelkan sebagai kumpulan titik-titik $(x, f(x))$. Nilai x menyatakan satuan waktu sedangkan $f(x)$ menyatakan nada yang dimainkan pada saat durasi melodi mencapai x . Misalnya, melodi pada Gambar 9 dapat ditransformasikan menjadi kumpulan titik-titik pada Gambar 10.



Gambar 9. Contoh Melodi



Gambar 10. Pemetaan Melodi ke Diagram Fungsi Waktu

Pada Gambar 10, sumbu x merepresentasikan waktu dalam satuan ketukan (pada gambar di atas, satu nada seperdelapan memiliki durasi satu ketukan), sedangkan sumbu y merepresentasikan interval nada yang sedang dimainkan dari nada dasar melodi tersebut. Ada beberapa aturan yang perlu diperhatikan dalam mengoversi sebuah melodi menjadi diagram fungsi interval terhadap waktu:

1. Panjang ketukan ditentukan dari *denominator* pada tanda birama. Misalnya, jika sebuah melodi dimainkan pada tanda birama 4/4, maka durasi satu ketukan pada melodi tersebut sama dengan durasi satu nada seperempat.
2. Interval antara nada pertama dengan nada dasar pada melodi tersebut berjarak sejauh-jauhnya 11 *semitone*. Dengan demikian, nilai *pitch* dari nada pertama selalu berkisar dari 0 sampai 11.

Pembandingan Melodi

Pada proses ini akan dilakukan penghitungan derajat ketidakmiripan antara melodi yang dijadikan *query* dengan melodi pada file MIDI. Algoritma untuk mencari melodi yang mirip dengan melodi M yang dijadikan *query* dengan panjang b birama adalah sebagai berikut:

1. Untuk setiap birama ke- i dalam melodi file MIDI, ambil melodi m yang merupakan potongan melodi dari file MIDI yang dimulai dari birama ke- i dan memiliki panjang b birama.

2. Nyatakan waktu awal m sebagai t_1 dan waktu akhir m sebagai t_2 . Nyatakan juga waktu awal M sebagai T_1 dan waktu akhir M sebagai T_2 . T_1 diberi nilai 0, sedangkan T_2 adalah jumlah ketukan antara nada pertama dengan nada terakhir M .
3. Tentukan fungsi $f(x)$ yang menyatakan interval nada dalam melodi m pada saat x terhadap nada dasar m . Tentukan juga fungsi $F(x)$ yang menyatakan interval nada dalam melodi M pada saat x terhadap nada dasar M .
4. Tentukan fungsi $g(x)$ yang merupakan hasil transformasi $f(x)$ dalam interval $[t_1, t_2]$ ke dalam interval $[T_1, T_2]$. Dengan demikian, $g(x) = f(t_1 + (x \cdot \frac{t_2 - t_1}{T_2 - T_1}))$
5. Tentukan derajat ketidakmiripan $F(x)$ dengan $g(x)$ pada interval $[T_1, T_2]$ dengan cara menghitung $L = S_1 + S_2$ di mana
 $S_1 = \sum \int_a^b (F(x) - g(x)) dx$ untuk tiap interval $[a, b]$ di mana $F(x) > g(x)$,
dan $S_2 = \sum \int_a^b (g(x) - F(x)) dx$ untuk tiap interval $[a, b]$ di mana $g(x) > F(x)$.
 L selalu bernilai positif. Jika nilai L semakin kecil, maka m semakin mirip dengan M . Jika $L = 0$, maka m sama dengan M .

Secara garis besar, *pseudocode* algoritma pembandingan melodi dapat dilihat pada Tabel 3.

Tabel 3. Algoritma Pembandingan Melodi

<p>M = melodi query dengan panjang b birama dan durasi $[T1 .. T2]$ m = melodi potongan dari file MIDI dengan panjang b birama dan durasi $[t1 .. t2]$ $F(x)$ = interval nada melodi M pada saat x $f(x)$ = interval nada melodi m pada saat x $g(x)$ = transformasi $f(x)$ dari $[t1 .. t2]$ ke $[T1 .. T2]$ Bandingkan $F(x)$ dengan $g(x)$ Return DerajatKetidakmiripan</p>

Berikut ini adalah contoh pembandingan dua buah melodi. Kedua buah melodi tersebut ditunjukkan pada Gambar 3.7 dan Gambar 3.8.

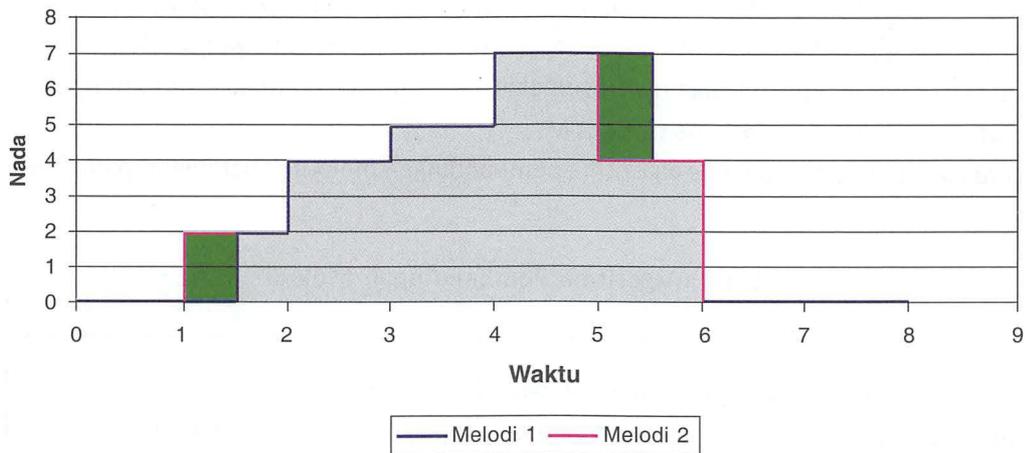


Gambar 10. Contoh Melodi 1



Gambar 11. Contoh Melodi 2

Jika kedua melodi di atas dipetakan ke diagram interval terhadap waktu, hasilnya akan tampak seperti pada Gambar 12.



Gambar 12. Pemetaan Melodi 1 dan Melodi 2 ke Diagram Koordinat

Dari Gambar 12, besar derajat ketidakmiripan dapat diperoleh dengan cara menghitung total luas *area* yang berwarna hijau.

Pengujian

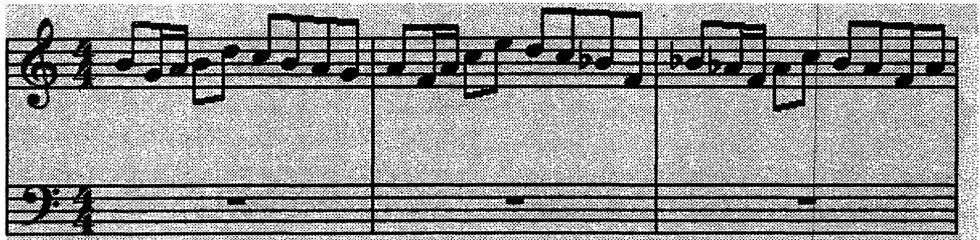
Tujuan dari pengujian ini adalah untuk membandingkan hasil pencarian melodi oleh perangkat lunak dengan hasil pencarian melodi oleh manusia. Pengujian dilakukan dengan cara memperdengarkan 7 file MIDI berisi lagu "Light My Fire" (yang diberi judul "Light My Fire (1).mid" hingga "Light My Fire (7).mid") kepada 25 orang responden. Melodi intro lagu "Light My Fire" dipilih untuk eksperimen ini karena unik, mudah diingat dan mudah dibedakan dari melodi lain yang serupa. Masing-masing responden diminta untuk memilih file MIDI manakah yang memiliki melodi intro yang paling mirip dengan versi aslinya dan file MIDI manakah yang memiliki melodi intro yang paling tidak mirip dengan versi aslinya. Melodi dari intro lagu "Light My Fire" pada tiap file MIDI yang diperdengarkan pada responden dapat dilihat pada Gambar 13 sampai Gambar 19.



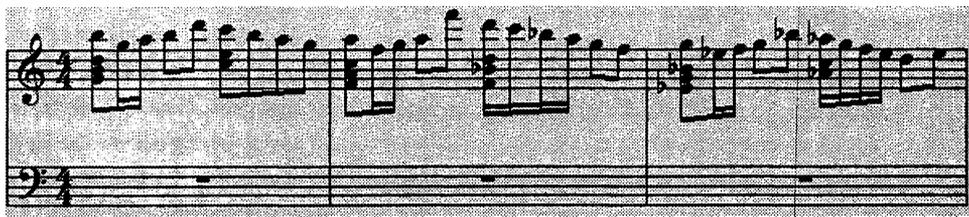
Gambar 13. Melodi Intro pada File "Light My Fire (1).mid"



Gambar 14. Melodi Intro pada File "Light My Fire (2).mid"



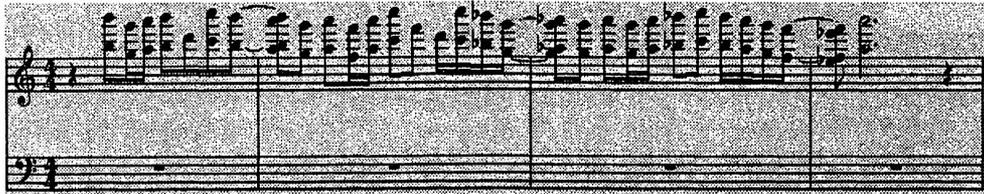
Gambar 15. Melodi Intro pada File "Light My Fire (3).mid"



Gambar 16. Melodi Intro pada File "Light My Fire (4).mid"



Gambar 17. Melodi Intro pada File "Light My Fire (5).mid"

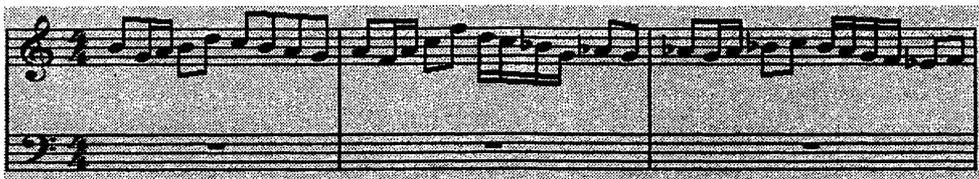


Gambar 18. Melodi Intro pada File "Light My Fire (6).mid"



Gambar 19. Melodi Intro pada File "Light My Fire (7).mid"

Melodi intro aslinya dapat dilihat pada Gambar 20.



Gambar 20. Melodi Intro "Light My Fire" Yang Asli

Hasil *polling* untuk melodi intro yang paling mirip dengan aslinya dapat dilihat pada Tabel 4.

Tabel 4.

Hasil *Polling* Untuk File MIDI Dengan Melodi Intro “Light My Fire” Yang Paling Mirip Dengan Aslinya

Nama File	Jumlah Responden
Light My Fire (1).mid	0
Light My Fire (2).mid	14
Light My Fire (3).mid	0
Light My Fire (4).mid	0
Light My Fire (5).mid	0
Light My Fire (6).mid	10
Light My Fire (7).mid	1

Hasil *polling* untuk melodi intro yang paling tidak mirip dengan aslinya dapat dilihat pada Tabel 5.

Tabel 5.

Hasil *Polling* Untuk File MIDI Dengan Melodi Intro “Light My Fire” Yang Paling Tidak Mirip Dengan Aslinya

Nama File	Jumlah Responden
Light My Fire (1).mid	2
Light My Fire (2).mid	0
Light My Fire (3).mid	23
Light My Fire (4).mid	0
Light My Fire (5).mid	0
Light My Fire (6).mid	0
Light My Fire (7).mid	0

Pencarian melodi dilakukan dengan memasukkan melodi pada Gambar 20 sebagai melodi *query*. Hasil dari pencarian melodi dengan perbandingan melodi per birama dapat dilihat pada Tabel 6.

Tabel 6.
Hasil Pencarian Melodi Yang Paling Mirip Dengan Melodi Intro "Light My Fire"
Dengan Perbandingan Melodi Per Birama

Nama File	Derajat Ketidakmiripan Minimum
Light My Fire (1).mid	12.5
Light My Fire (2).mid	0
Light My Fire (3).mid	17.65
Light My Fire (4).mid	12.1
Light My Fire (5).mid	0.75
Light My Fire (6).mid	33.25
Light My Fire (7).mid	0.5

Dari Tabel 6, terlihat bahwa melodi intro pada file "Light My Fire (2).mid" sama persis dengan melodi intro aslinya. Sebenarnya melodi intro pada file "Light My Fire (6)" juga sama persis dengan intro aslinya, tapi karena melodi pada file "Light My Fire (6)" dimulai dari ketukan ke-2 sedangkan melodi pada *query* dimulai dari ketukan pertama, maka perbandingan kedua melodi tersebut menghasilkan derajat ketidakmiripan yang paling besar di antara file-file MIDI lainnya. Hasil dari pencarian melodi dengan perbandingan melodi per nada dapat dilihat pada Tabel 7.

Tabel 7.
Hasil Pencarian Melodi Yang Paling Mirip Dengan Melodi Intro "Light My Fire"
Dengan Perbandingan Melodi Per Nada

Nama File	Derajat Ketidakmiripan Minimum
Light My Fire (1).mid	12.5
Light My Fire (2).mid	0
Light My Fire (3).mid	15.69
Light My Fire (4).mid	12.1
Light My Fire (5).mid	0.75
Light My Fire (6).mid	0
Light My Fire (7).mid	0.5

Dari Tabel 7, terlihat bahwa melodi intro pada file "Light My Fire (2).mid" dan "Light My Fire (6).mid" sama persis dengan melodi intro aslinya. Pencarian terhadap file "Light My Fire (6).mid" bisa mengembalikan derajat ketidakmiripan minimum nol karena perbandingan melodi dilakukan per nada. Kelemahan perbandingan melodi per nada jika dibandingkan dengan perbandingan melodi per birama adalah waktu eksekusinya yang bisa mencapai puluhan kali lebih lama.

PENUTUP

Kesimpulan

Berdasarkan hasil pengujian, ada beberapa hal yang dapat disimpulkan, antara lain:

1. Metode-metode konversi melodi polifonik ke monofonik yang digunakan menghasilkan melodi monofonik yang cukup representatif untuk digunakan dalam pencarian melodi.

2. Metode perbandingan melodi per birama sangat sensitif terhadap posisi nada dalam ketukan. Jika ada dua melodi yang sama persis tapi dimainkan pada ketukan yang berbeda, maka perbandingan kedua melodi tersebut dengan metode ini dapat menghasilkan derajat ketidakmiripan yang besar.
3. Metode perbandingan melodi per nada dapat menghasilkan keluaran yang lebih akurat daripada metode perbandingan melodi per birama tapi membutuhkan waktu puluhan kali lebih lama.

DAFTAR PUSTAKA

1. Yongwei Zhu, M. Kankanhalli, Qi Tian. 2002. *Similarity Matching Of Continuous Melody Contours For Humming Querying Of Melody Databases*. National University Of Singapore.
2. Hung-Ming Yu, Wei Ho Tsai, Hsin-Min Wang. 2006. *A Music Retrieval System Based On Query-by-singing For Karaoke Jukebox*. Academica Sinica, Taiwan.
3. Alexandra L. Uitdenboger, Justin Zobel. 1998. *Manipulation Of Music For Melody Matching*. Computer Science Department, Royal Melbourne Institute Of Technology, Melbourne, Australia.
4. Iman S. H. Suyoto, Alexandra L. Uitdenboger. 2004. *Exploring Microtonal Matching*. School Of Computer Science And Information Technology, RMIT University.
5. Benoit Meudic. 2003. *Musical Similarity In A Polyphonic Context : A Model Outside Time*. Ircam, Paris.
6. Maureen Gunawan. 1999. *Pembangkit Harmonisasi Chord*. Jurusan Teknik Informatika, ITB.
7. J. Stephen Downie. 2003. *Music Information Retrieval*. University Of Illinois.
8. *Themefinder*. Situs Web. <http://www.themefinder.org>
9. *Meldex*. Situs Web. <http://www.sadl.uleth.ca/nz/cgi-bin/music>
10. *MIDI File Format*. 2002. Situs Web. <http://www.chordwizard.com>
11. Avery Wang. 2006. *The Shazam Music Recognition Service*. *Majalah Communications* Volume 49.
12. Stephen Downie, Michael Nelson. 2005. *Evaluation Of A Simple And Effective Music Information Retrieval Method*. University Of Illinois, University Of Western Ontario.
13. S. Doraisamy, G. Widmer. 2004. *A Polyphonic Music Retrieval System Using N-grams*.
14. A. Ghias, J. Logan, D. Chamberlin, B.C. Smith. 1995. *Query By Humming: Musical Information Retrieval In An Audio Database*.
15. Mathias Robine, Pierre Hanna, Pascal Ferraro. 2007. *Music Similarity: Improvements Of Edit-based Algorithms By Considering Music Theory*. Université Bordeaux.