

Pengaplikasian *Binary Tree* untuk Mengecek Kesahihan suatu *Sheet Music*

Leonardo / 13517048
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517048@std.stei.itb.ac.id

Abstract—Meskipun tidak mengerti cara membacanya, anda pasti setidaknya pernah melihat *sheet music*. *Sheet music* sangat berguna untuk musisi dalam menuliskan apa yang telah atau akan dimainkannya dalam bentuk *output* audio. Tetapi bagaimana cara musisi mengetahui apakah *sheet music* yang ia baca sah atau tidak? Pada Makalah ini, akan dijelaskan apa itu *sheet music*, bagaimana menghubungkannya dengan teori *Binary Tree*, dan permasalahan apa yang dapat diselesaikan dengan teori tersebut. Pada makalah ini juga akan dicantumkan hasil-hasil analisis yang didapatkan melalui metode *Binary Tree* ini.

Keywords—*Binary tree*, *Sheet music*, Ketukan, *Bar*

I. PENDAHULUAN

Pada jaman modern ini, sudah menjadi salah satu kriteria dasar untuk musisi-musisi pada genre apapun untuk setidaknya dapat membaca *sheet music*, tetapi untuk membaca dengan benar, para musisi juga harus dapat mengecek apakah *sheet music* yang ia baca *valid* atau tidak. Dapat dilihat dari *bar* berikut,



Gambar 1.1 suatu *bar* sederhana dalam *sheet music* (aplikasi *Muscore*, dibuat pada 7 Desember 2018, pukul 11.55)

Mungkin dengan penjumlahan ketukan sederhana, bisa didapatkan apakah *bar* tersebut memiliki jumlah ketukan yang sah atau tidak. Akan tetapi, pada era klasikal (1775-1825)^[1], *sheet music* yang ada tidak semudah itu dibaca dan cenderung sulit untuk dimengerti musisi-musisi muda. Salah satu *bar*-nya dapat bertuliskan



Gambar 1.2 suatu *bar* rumit dalam *Flute Concerto in G major K.313/285c (Mozart, Wolfgang Amadeus) mov. 2 Adagio ma non troppo* (milik sendiri, diakses pada 7 Desember 2018, pukul 12.09)

akan sangat sulit untuk membaca *bar* tersebut hanya dengan

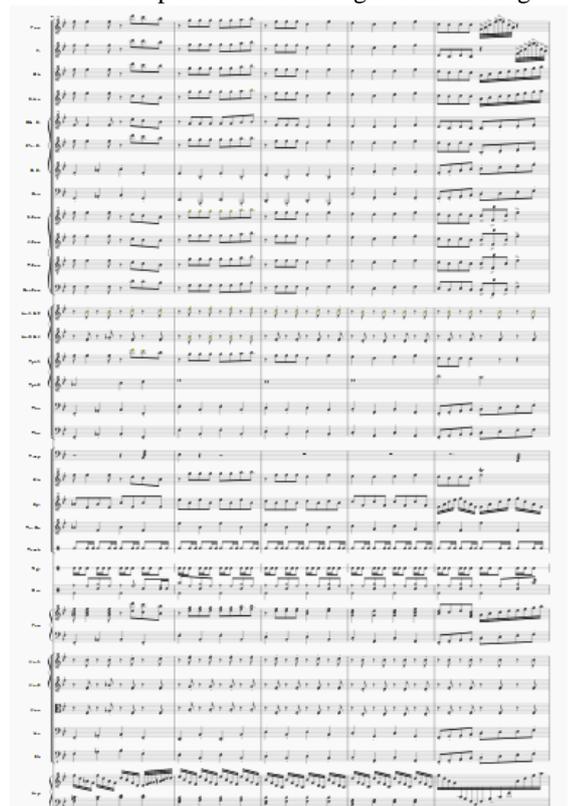
metode penjumlahan biasa. Oleh sebab itu, digunakan metode lain yang menggunakan *binary tree*.

II. DASAR TEORI

Pada bagian ini akan dijelaskan apa itu *sheet music*, apa itu *sheet music* yang sah, apa itu pohon dalam ilmu matematika, dan apa itu *binary tree*.

A. *Sheet Music*

Sheet music adalah suatu bentuk fisik lagu berbentuk notasi-notasi musik yang menjelaskan ritme, akord, nada, dan juga ketukan suatu lagu dengan menggunakan *bar* dan not balok. Setiap not balok diletakkan pada suatu sistem garis dan lubang.



Gambar 2.1 suatu halaman dari *sheet music* orkestral *Hitorigoto (Clarinet)* (aplikasi *Muscore*, milik sendiri, diakses pada 7 Desember 2018, pukul 12.27)

Pada suatu *sheet music*, terdapat istilah bernama *clef*. *Clef* merupakan symbol musikal yang menentukan pada nada dasar apa lagu tersebut dimainkan. Setiap *clef* memiliki suatu nada *pivot*. Untuk dapat menggambar *clef* dengan tangan pada suatu *sheet music*, menggambar selalu dimulai dari nada *pivot* tersebut. Setiap alat musik memakai *clef* yang tidak tentu sama dengan alat musik-alat musik lainnya dan terkadang suatu alat musik dapat memakai lebih dari satu *clef* (contoh : piano). **Setiap garis dan lubang yang berbeda pada *clef* yang berbeda mengartikan nada yang berbeda.** Berikut merupakan contoh-contoh *clef*,

a. *Treble Clef*

Treble clef merupakan jenis *clef* yang paling umum. *Clef* ini biasa digunakan untuk alat-alat bernada tinggi, berfrekuensi tinggi, dan alat-alat yang memainkan bagian vokal pada suatu lagu. Nada *pivot* untuk *treble clef* adalah G4. Maka dari itu, *treble clef* sering juga disebut ‘kunci G’.

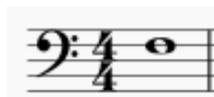


Gambar 2.2 Treble clef dan nada G4 (aplikasi Musescore, dibuat pada 7 Desember 2018, pukul 20.41)

Alat-alat musik yang biasa menggunakan *treble clef* adalah flute, klarinet, tangan kanan piano, penyanyi (soprano, alto, dan tenor) dan juga biola.

b. *Bass Clef*

Bass clef merupakan jenis *clef* yang paling sering digunakan untuk alat-alat bernada rendah dan cenderung berfrekuensi rendah. Nada *pivot* untuk *bass clef* adalah F3. Maka dari itu, *bass clef* sering juga disebut ‘kunci F’.



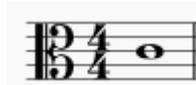
Gambar 2.3 Bass clef dan nada F3 (aplikasi Musescore, dibuat pada 7 Desember 2018, pukul 20.47)

Alat-alat musik yang biasa menggunakan *bass clef* adalah *double bass*, bassoon, tangan kiri piano, penyanyi bass, dan juga *cello*.

c. *Alto Clef*

Alto clef merupakan *clef* yang paling jarang digunakan. *Clef* ini digunakan pada umumnya untuk alat-alat bernada tengah dan cenderung memiliki bunyi yang tersembunyi pada suatu orkestra. Nada *pivot* untuk *alto clef* adalah C4. Maka dari itu, *alto clef* sering

juga disebut ‘kunci C’.



Gambar 2.4 Alto clef dan nada C4 (aplikasi Musescore, dibuat pada 7 Desember 2018, pukul 20.54)

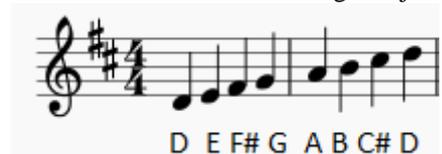
Alat-alat musik yang biasa menggunakan *alto clef* adalah *viola* dan *alto trombone*. Alat lain yang menggunakan *alto clef* biasanya tidak menggunakan *alto clef* sebagai kunci utamanya.

Pada makalah ini, tidak akan dijelaskan semua jenis *clef* yang ada, akan tetapi istilah *clef* akan sangat banyak digunakan pada makalah ini karena suatu *clef* selalu diletakkan pada awal baris di setiap *sheet music* dan dijadikan acuan untuk letak objek-objek yang ingin ditinjau.

Pada *sheet music* juga terdapat info-info berupa nada, ketukan, dinamika, dll. Pada makalah ini, hanya akan dijelaskan nada dan ketukan saja, karena info-info yang lain dirasa tidak relevan.

1. Nada

Nada di dalam suatu *sheet music* menjelaskan *note* apa yang perlu dimainkan pada saat sampai ke nada tersebut. Nada tersebut dimainkan pada suatu tangga nada (*Key Signature*) yang spesifik untuk setiap lagu yang ada. Dan tangga nada pada suatu lagu dapat dimodulasi atau diubah-ubah dari waktu ke waktu selama lagu berjalan.



Gambar 2.5 Tangga nada D mayor (aplikasi Musescore, dibuat 7 Desember 2018, pukul 12.30)

Pada suatu nada-nada khusus, terdapat notasi (#) yang merupakan tanda *sharp*, dan (b) yang merupakan tanda *flat*.

Informasi mengenai tangga nada suatu lagu dijelaskan pada bagian kiri halaman, di kanan *clef*. Untuk mengubah-ngubah tangga nada, digunakan kembali notasi *sharp*(#) dan *flat*(b)^[2].

Dengan mengidentifikasi letak garis manakah yang memiliki *sharp* atau *flat*, kita memainkan nada yang ada pada garis tersebut (dan oktaf atas bawahnya) seperti pada gambar 2.2, terdapat *sharp* pada garis teratas (F) dan lubang kedua teratas (C), maka musisi memainkan nada F menjadi F#, dan C menjadi C#, membentuk

akord D mayor.

Berikut akan dijelaskan lebih jelas tentang apa itu *sharp* dan *flat*.

- *Sharp* (#)
Sharp (disebut juga *cres*) digunakan untuk 'menaikkan' suatu nada sebanyak $\frac{1}{2}$.

Pada suatu lagu, tangga nada dapat diidentifikasi dengan banyaknya *sharp* yang ada pada kanan *clef*. Penjelasananya adalah sebagai berikut,

- 1 (G mayor)
- 2 (D mayor)
- 3 (A mayor)
- 4 (E mayor)
- 5 (B mayor)
- 6 (F# mayor)
- 7 (C# mayor)

- *Flat* (b)
Flat (atau *mol*) digunakan untuk 'menurunkan' suatu nada sebanyak $\frac{1}{2}$.

Pada suatu lagu, tangga nada dapat diidentifikasi dengan banyaknya *flat* yang ada pada kanan *clef*. Penjelasananya adalah sebagai berikut,

- 1 (F mayor)
- 2 (B \flat mayor)
- 3 (E \flat mayor)
- 4 (A \flat mayor)
- 5 (D \flat mayor)
- 6 (G \flat mayor)
- 7 (C \flat mayor)

Untuk tangga nada C mayor, penanda tangga nada tidak bertuliskan apa-apa, sehingga disebut juga tangga nada *natural*(\natural).

Perlu dicatat bahwa apabila ada suatu nada yang telah diberikan *sharp* atau *flat* pada suatu *bar*, apabila di *bar* yang sama, nada yang tadinya diubah dipanggil kembali, maka nada tersebut adalah nada yang sudah *disharp* atau *diflat*-kan. Terkecuali apabila nada tersebut dipanggil kembali dengan notasi *natural*(\natural).

2. Ketukan

Ketukan di dalam suatu *sheet music* menjelaskan seberapa lama (ketergantungan tempo) musisi harus memainkan suatu not (termasuk *rest note*).

Pada suatu *bar*, terdapat beberapa ketukan, dan

berapa ketukan yang ada pada suatu *bar* ditentukan oleh angka disamping kanan *clef*. Penanda itu disebut *time signature*. Terdapat beberapa jenis *time signature*, salah satu yang paling umum adalah 4/4, 2/4, 2/2, 3/4, dan 6/8. Pada setiap lagu, *time signature* dideklarasikan pada *bar* pertama lagu dan pada setiap ada pergantian *time signature* itu sendiri.

Time signature 4/4 berarti memiliki setiap *bar* memiliki ukuran 4 ketukan *quarter note*. Contoh lain adalah *time signature* 6/8 yang berarti setiap *bar* memiliki ukuran 6 ketukan *eighth note*.

Pada makalah ini, hanya akan digunakan 4/4 dan 6/8 karena kedua *time signature* tersebut merupakan *time signature* yang paling mudah diidentifikasi dan yang paling sering digunakan di dunia nyata.



Gambar 2.6 Jenis-jenis ketukan pada nada C (4/4) (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 12.38)

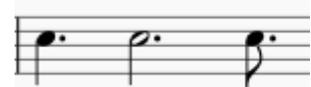


Gambar 2.7 Jenis-jenis ketukan *rest note* (4/4) (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 12.40)

Dari gambar 2.6 dan 2.7 dapat dijelaskan bahwa ada banyak jenis ketukan dasar. pada setiap *bar*-nya untuk mempermudah penjelasan.

- Bar 1 : *Whole Note* (berukuran 4 ketuk)
- Bar 2 : *Half Note* (berukuran 2 ketuk)
- Bar 3 : *Quarter Note* (berukuran 1 ketuk)
- Bar 4 : *Eighth Note* (berukuran $\frac{1}{2}$ ketuk)
- Bar 5 : *16th Note* (berukuran $\frac{1}{4}$ ketuk)
- Bar 6 : *32nd Note* (berukuran $\frac{1}{8}$ ketuk)

Adapula jenis nada berupa *dotted note*. *Dotted note* merupakan nada yang memiliki 1,5 kali ukuran ketukannya sendiri.



Gambar 2.8 segelintir jenis *dotted note*. (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 12.45)

Penjelasan gambar :

- kiri : *dotted quarter note* (berukuran $1\frac{1}{2}$ ketuk)
- tengah : *dotted half note* (berukuran 3

- ketuk)
- kanan : *dotted eighth note* (berukuran $\frac{3}{4}$ ketuk)

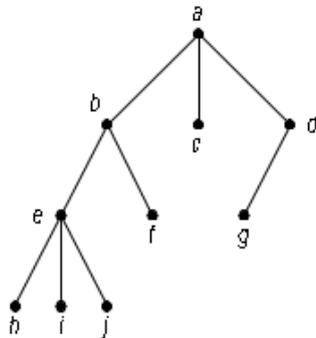
Masih banyak jenis ketukan, akan tetapi ketukan-ketukan yang dijelaskan sebelumnya merupakan ketukan-ketukan yang pada umumnya mudah untuk ditemukan dimana-mana.

Pada suatu *sheet music*, umumnya terdapat kombinasi dari semua atau paling tidak sebagian besar dari nada dan ketukan yang dijelaskan sebelumnya untuk membuat lagu yang utuh dan lengkap.

Suatu *sheet music* terdiri dari satu atau lebih *bar*. Setiap *bar* ini memiliki kumpulan nada-nada dalam berbagai ketukan. Suatu *sheet music* dikatakan *valid* jika dan hanya jika semua *bar* yang ada pada *sheet music* tersebut *valid*. Suatu *bar* dinyatakan *valid* atau sah jika dan hanya jika jumlah ketukan di dalam *bar* tersebut sama jumlahnya dengan jumlah ketukan yang ditentukan oleh *time signature* dalam bagian lagu tersebut. Apabila suatu *sheet music* memiliki satu saja *bar* yang tidak sah, maka *sheet music* tersebut tidaklah sah.

B. Pohon

Pohon (dalam ilmu matematika) merupakan suatu graf khusus yang tidak memiliki sisi ganda, sisi gelang, dan juga sirkuit (yang berarti tidak ada jalan dari suatu *node* ke *node* itu sendiri melalui *node-node* yang lain). Pada suatu pohon, jalan antara 1 *node* ke *node* lain hanya memiliki 1 buah jalan yang unik.



Gambar 2.9 Contoh Pohon (sumber : [3])

Suatu pohon memiliki beberapa terminologi, yaitu

1. *N-ary*
Suatu pohon dinyatakan *n-ary* apabila pohon tersebut pada setiap *node*-nya memiliki maksimum *n* buah *child node*.
2. *Node*
Node dari suatu tree adalah sebuah simpul di dalam tree itu sendiri. *Root node* merupakan *node* yang tidak memiliki *parent* dan memiliki *level* terendah dari suatu pohon. Sementara *leaf node* merupakan *node* yang tidak

memiliki *child* dan memiliki *level* tertinggi dari suatu pohon.

3. *Child* dan *Parent*

Suatu *node* B dikatakan sebagai *child* atau anak dari *node* A jika dan hanya jika ada 1 buah sisi yang menghubungkan antara kedua *node* tersebut dan *node* A ada pada satu *level* di bawah *node* B, dan oleh karena itu, *node* A disebut sebagai *parent* atau orang tua dari *node* B.

4. *Sibling*

Suatu *node* A dikatakan sebagai *sibling* dari *node* B jika dan hanya jika kedua buah *node* merupakan *child* dari *parent* yang sama dan kedua buah *node* berada pada *level* yang sama.

5. *Descendant* dan *Ancestor*

Suatu *node* A dikatakan sebagai *descendant* atau keturunan dari *node* B jika dan hanya jika ada *path* di antara *node* A dan *node* B dan *node* B memiliki *level* yang lebih tinggi dari *node* A, dan oleh karena itu, *node* B merupakan *ancestor* atau leluhur dari *node* A.

6. *Path*

Lintasan atau *path* dari suatu *node* A ke *node* B adalah *node-node* lain yang dilewati dari *node* A untuk mencapai ke *node* B.

7. *Level*

Level dari suatu *node* adalah 1 apabila ia merupakan *root node*. *Level* dari *node* lain merupakan *level* dari *parent node*-nya ditambah dengan 1.

8. *Height*

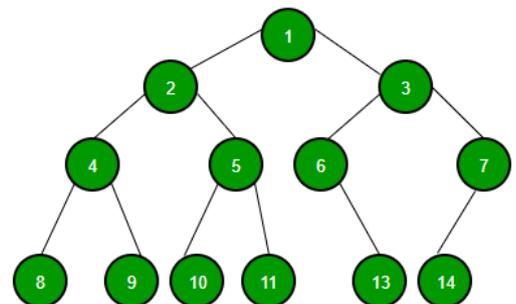
Tinggi atau *height* dari suatu pohon merupakan *level* maksimum dari suatu pohon. [4]

9. *Sub-Tree*

Sub-Tree atau upapohon dari suatu pohon A merupakan pohon B yang memiliki *root node* yang sama dengan salah satu *node* pada pohon A dan memiliki semua *descendant node* dari *root node* tersebut.

C. *Binary Tree*

Binary tree (atau pohon biner) merupakan suatu jenis pohon *n-ary* dengan $n = 2$ yang setiap *parent node*-nya memiliki maksimal 2 buah *child node*.



Gambar 2.10 Contoh *Binary Tree* (sumber : [5])

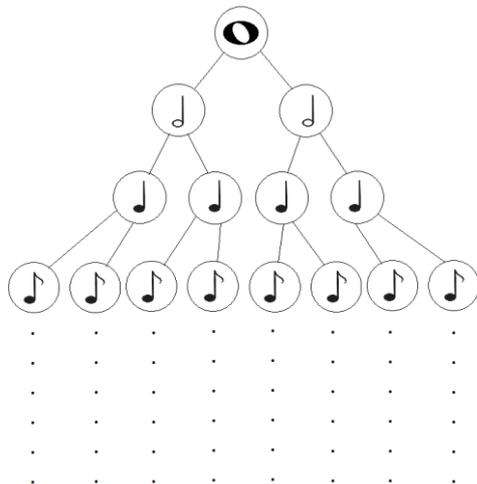
Binary Tree sebagai suatu struktur data memiliki 3 buah komponen, info *node*, alamat *left child*, dan alamat *right child*^[3].

Suatu pohon dinyatakan sebagai *balanced tree* (atau pohon seimbang) apabila pohon tersebut memiliki upapohon-upapohon yang seimbang dan tinggi dari masing-masing upapohon adalah sama atau hanya memiliki perbedaan 1 elemen.

III. METODE PEMECAHAN MASALAH

Pada pengaplikasiannya, pengecekan akan dilakukan dalam satu-per-satu *bar*, untuk satu buah *bar*, dibuat satu *binary tree*. *Binary tree* tersebut dibuat sedemikian sehingga agar *root node*-nya merupakan jumlah ketukan hasil penjumlahan dari kedua *child node*-nya, dan *leaf node*-nya merupakan semua nada dengan ketukannya. Apabila terbentuk nada yang merepresentasikan banyaknya ketukan pada *bar* tersebut, *bar* tersebut dinyatakan sahih. Akan tetapi apabila nada yang terbentuk bukanlah nada yang merepresentasikan banyaknya ketukan pada *bar* tersebut, maka *bar* tersebut dinyatakan tidak sah.

Dalam penggunaannya, kita memiliki beberapa bentuk umum. Bentuk-bentuk umum tersebut dibedakan berdasarkan *time signature* dari lagu tersebut.



Gambar 3.1 Contoh bentuk umum *binary tree* untuk lagu 4/4 (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 19.04)

Penjelasan akan gambar :

- Nada teratas merupakan total banyaknya nada yang terbentuk dari kumpulan nada-nada pada suatu *bar*. Dalam hal ini, didapat *whole note*.
- *Whole note* memiliki 2 buah *child node* yang merupakan 2 *half note*. Seperti yang telah dijelaskan pada bab 2 di bagian ketukan, *half note* berukuran 2 ketuk. Sementara *whole note* berukuran 4 ketuk. Setiap *parent node* memiliki ukuran dua kali ukuran

anakanya.

- *Sibling node* selalu diusahakan untuk memiliki ukuran ketukan yang sama.
- Titik-titik pada gambar menyatakan bahwa di bawah *eighth node* masih mungkin untuk menambahkan *child node-child node* yang lain.

Binary Tree yang penulis gunakan disini bukanlah pohon seimbang. Pada penggunaannya, nada yang ada pada suatu bar nantinya akan diletakkan pada *leaf node* sesuai dengan dimainkannya pada bar tersebut (dari kiri ke kanan).

IV. HASIL PENELITIAN DAN ANALISIS

Pada bagian ini, penulis akan menyelesaikan beberapa masalah dengan metode yang telah dijelaskan pada bagian 3 makalah ini. Masalah-masalah yang diselesaikan memiliki berbagai jenis variasi, penulis juga akan mencantumkan contoh masalah yang unik, dan juga yang tidak sah. Terdapat 5 buah contoh penyelesaian masalah pada makalah ini.

Contoh 1, Contoh *bar* sederhana dasar

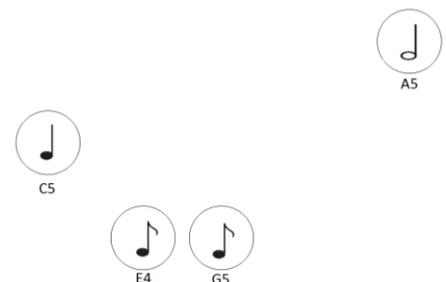


Gambar 4.1 Contoh 1, contoh *bar* sederhana dasar (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 21.10)

Pada contoh ini, penulis menggunakan contoh sederhana yang juga dapat dilihat pada bagian pendahuluan.

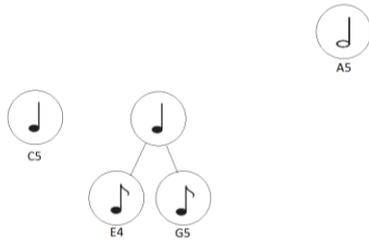
Pada contoh *bar* ini, terdapat 4 buah nada, 1 buah *quarter note*, 2 buah *eighth note*, dan 1 buah *half note*. *Time signature* pada bar tersebut merupakan 4/4 seperti yang dapat dilihat di kanan *clef* pada gambar 4.1. Pada contoh 1 ini, penulis akan memberikan langkah-langkah kerja yang harus dilakukan untuk menyelesaikan permasalahan ini. Pada contoh-contoh berikutnya, penulis tidak akan memberikan langkah-langkah pengerjaan.

Langkah 1 : Buat *leaf node* sebanyak nada yang ada pada bar tersebut, lalu sesuaikan tingginya berdasarkan ukuran dari nada tersebut, cantumkan pula nada apa yang ada pada *leaf node* tersebut.



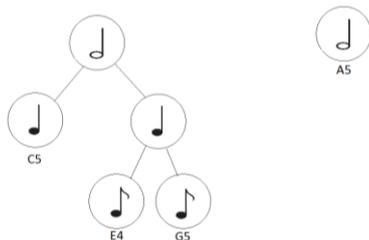
Gambar 4.2 *leaf nodes* dari bar di gambar 4.1 (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 21.11)

Langkah 2 : Buat *parent node* dari 2 *node* yang bersampingan (diusahakan yang memiliki 2 ukuran sama terlebih dahulu) dan menjadikannya nada dengan ukuran 1 lebih atas dari *child node*-nya.

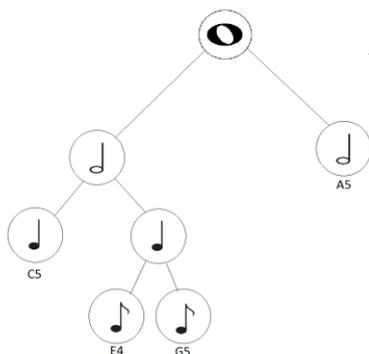


Gambar 4.3 pembuatan *node quarter note* dari dua buah *eighth note* (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 21.14)

Langkah 3 : Ulangi langkah 2 hingga terbentuk hanya 1 buah *node* yang merupakan *root node*.



Gambar 4.4 Pengulangan 1 (pembuatan *node half note* dari dua buah *quarter note*) (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 21.18)



Gambar 4.5 Pengulangan 2 (pembuatan *whole note* dari dua buah *half note*) (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 21.21)

Langkah 4 : Lakukan pengecekan apakah nada terakhir yang terbuat pada *binary tree* dan *bar* yang ingin dicek. Apabila kedua gambar memiliki ukuran ketukan yang sama, maka *bar* yang ingin dicek merupakan *bar* yang *valid* atau *sahih*.

Pada contoh 1 ini, *root node* dari pohon di gambar 4.5

adalah suatu *whole note*, yang berukuran 4 ketuk, pada gambar 4.1 kita dapat melihat bahwa *time signature*-nya adalah 4/4, yang berukuran 4. Maka *bar* contoh 1 merupakan *bar* yang *sahih*.

Contoh 2, Contoh dengan ketukan yang kurang jelas



Gambar 4.6 Contoh 2, contoh dengan ketukan yang kurang jelas (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 21.37)

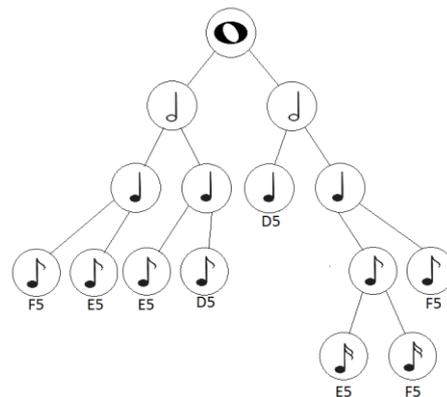
Pada contoh ini, kita dapat melihat nada dengan ketukan yang tidak lazim. pada contoh ini, kita harus mengubah bentuk dari nada-nada ini menjadi



Gambar 4.7 Bar contoh 2 setelah diubah (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 21.42)

Perlu diperhatikan bahwa gambar 4.6 dan gambar 4.7 memiliki *output* suara yang sama saat dimainkan oleh musisi, akan tetapi ketukan pada gambar 4.7 lebih mudah dibaca karena ketukan diletakkan lebih jelas dan lebih rapi.

Setelah mengubah bentuk dari *bar* tersebut, kita dapat melakukan 4 langkah yang penulis lakukan pada contoh 1. Setelah melakukan 4 langkah tersebut, didapat *binary tree* berikut,



Gambar 4.8 Hasil pembuatan *binary tree* dari gambar 4.7 (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 21.52)

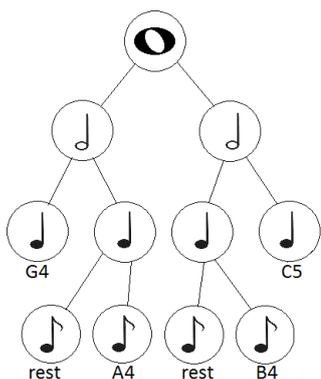
Seperti yang kita bisa lihat dari gambar 4.8, *root node* pada pohon di gambar 4.8 adalah suatu *whole note*. Pada gambar 4.6 dan 4.7 diketahui bahwa *time signature*-nya adalah 4/4, maka *bar* contoh 2 adalah *bar* yang *sahih*.

Contoh 3, Contoh bar yang memiliki *rest notes*



Gambar 4.9 Contoh 3, contoh dengan *rest notes* (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 22.01)

Pada contoh ini, dapat dilihat ada 2 *eighth rest note*. *Rest note* diperlakukan sama dengan *note* lainnya, oleh karena itu, dihasilkan *binary tree* berikut,



Gambar 4.10 Hasil pembuatan *binary tree* dari contoh 3 (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 22.08)

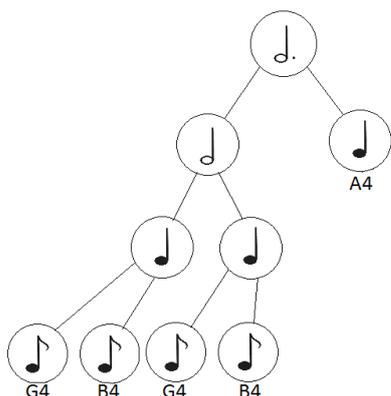
Pada gambar 4.10, terbentuk *whole note* sebagai *root node* dari pohon tersebut, karena gambar 4.9 menunjukkan bahwa *time signature*-nya 4/4, maka *bar* pada gambar 4.9 sahih.

Contoh 4, Contoh *bar* yang *invalid*



Gambar 4.11 Contoh 4, contoh *bar* yang *invalid* (aplikasi *Musescore* dan *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 22.11)

Pada contoh 4 ini, *binary tree* yang terbentuk adalah



Gambar 4.12 *Binary tree* dari contoh 4 (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 22.17)

Seperti yang bisa kita lihat pada *binary tree* di gambar 4.12, *root node* yang terbentuk adalah *dotted half note* yang berukuran 3 ketukan. Sementara *time signature* pada gambar 4.11 adalah 4/4 yang berukuran 4 ketuk. Karena *root node* dan *time signature*-nya berbeda, *bar* pada contoh gambar 4.11 *invalid* atau tidak sah.

Contoh 5, Contoh dengan menggunakan *time signature* 6/8



Gambar 4.13 Contoh 5, contoh dengan menggunakan *time signature* 6/8 (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 22.36)

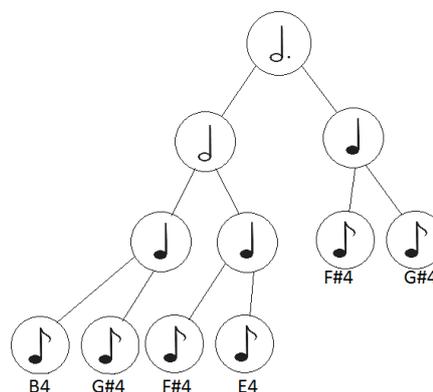
Pada samping kanan dari *treble clef*, dapat dilihat bahwa *time signature* yang dipakai disini merupakan *time signature* 6/8 yang berukuran 3 ketukan.

Pertama-tama perlu kita ubah bentuk pergarisannya menjadi 3 buah pasang *eighth note*. *Bar* pada contoh 5 setelah diubah berbentuk seperti berikut,



Gambar 4.14 Hasil pengubahan *bar* contoh 5 (aplikasi *Musescore*, dibuat pada 7 Desember 2018, pukul 22.42)

Pembentukan *binary tree* untuk gambar 4.14 adalah



Gambar 4.15 *Binary tree* dari contoh 5 (aplikasi *Microsoft Paint*, dibuat pada 7 Desember 2018, pukul 22.52)

Dari *binary tree* yang ada pada gambar 4.15, didapat *root node* yang berisi *dotted half note* yang berukuran 3 ketuk. Sementara *time signature* pada gambar 4.13 adalah 6/8 yang berukuran 3 ketuk. Maka *bar* pada gambar 4.13 merupakan *bar* yang sah.

V. KONKLUSI

Memang ada banyak cara di dunia ini untuk melakukan pengecekan kesahihan suatu *sheet music*, salah satu cara yang adalah dengan pembuatan *binary tree*. Dengan melakukan metode-metode pembuatan *binary tree* yang dilakukan pada bab 4 diatas, proses pengecekan menjadi lebih teliti dan lebih akurat. Pada makalah ini dicontohkan bagaimana melakukan pengecekan terhadap suatu *bar* di dalam suatu *sheet music*. Dengan metode yang sama, tidak hanya satu *bar* saja, namun satu *sheet music*-pun dapat dicek kesahihannya secara bertahap. Penulis harap makalah ini dapat berguna untuk masyarakat ke depannya serta metode ini dapat digunakan untuk mempermudah musisi-musisi di luar sana untuk melakukan pengecekan terhadap sah atau tidaknya suatu *sheet music*.

VI. UCAPAN TERIMA KASIH

Ucapan terima kasih saya berikan sebesar-besarnya kepada Tuhan Yang Maha Esa, yang telah memberkati pembuatan makalah ini sejak saya mulai mencari ide hingga makalah ini selesai. Terima kasih juga saya ucapkan kepada keluarga dan teman-teman saya yang telah mendoakan dan mendukung saya sampai sekarang.

Saya juga mengucapkan terima kasih yang sebesar-besarnya kepada Dr. Judhi Santoso, M.Sc. selaku dosen kelas saya untuk mata kuliah IF2120 Matematika Diskrit. Terima kasih juga saya ucapkan kepada Dr. Ir. Rinaldi Munir, MT. sebagai pemberi tugas makalah ini karena telah memberikan kami seangkatan sebuah kesempatan untuk mengasah kemampuan kami untuk membuat makalah ilmiah pada rumpun informatika.

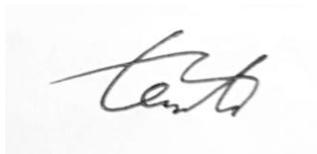
REFERENSI

- [1] <http://cmed.faculty.ku.edu/private/classical.html>, diakses pada 7 Desember 2018, pukul 11.56
- [2] <https://www.piano-keyboard-guide.com/the-difference-between-sharp-and-flat.html>, diakses pada 7 Desember 2018, pukul 15.19
- [3] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf), diakses pada 7 Desember 2018, pukul 16.25
- [4] https://www.sqa.org.uk/e-learning/LinkedDS04CD/page_05.htm, diakses pada 7 Desember 2018, pukul 16.28
- [5] <https://www.geeksforgeeks.org/binary-tree-data-structure/>, diakses pada 7 Desember 2018, pukul 16.30

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, 8 Desember 2018



Leonardo / 13517048