

Pengembangan Aplikasi Konversi Representasi Not Balok Ke Not Angka Untuk Paduan Suara Campur

Lukas Chrisantyo¹
lukaschris@mti.ugm.ac.id

Rudy Hartanto²
rudi@mti.ugm.ac.id

Lukito Edi Nugroho³
lukito@mti.ugm.ac.id

Abstract

Beam notation is a notation that is officially used in the whole world as an intermediary to do musical communication. In Indonesia, notation that is more absorbed and mostly easy to learn autodidactly is numbered notation, while beam notation is more difficult to learn and tend to be avoided. Traditionally, beam notation can be converted to numbered notation by fully relying on human skills and abilities. It is obvious that the conversion process would require much time and energy. Therefore, a new conversion application is developed to cope the conversion from beam notation into numbered notation specially for mixed choir, with represented in a XML file called MusicXML. MusicXML is a representation of beam notation that are more easily to be compressed and has a universal characteristic, can be opened in a wide range of applications. The purpose of this research was to design a fast, reliable and accurate application to convert beam notation into numbered notation for mixed choir which consist of multiple voice parts, according to the prevailing theory of music.

Kata-kunci: *rekayasa perangkat lunak, konversi notasi musik, MusicXML*

1. PENDAHULUAN

Not balok adalah standar penulisan notasi musik yang diakui di seluruh dunia sejak ratusan tahun silam hingga hari ini. Namun untuk bisa menguasainya dibutuhkan proses pembelajaran yang cukup panjang. Di Indonesia, orang-orang cenderung lebih terbiasa dan nyaman menggunakan not angka dalam membaca dan menulis notasi vokal. Dengan adanya perbedaan tata cara penulisan ini maka diperlukan proses konversi. Sampai penelitian ini dibuat, belum ditemukan satupun aplikasi konversi not balok ke not angka. Oleh karena itu dikembangkan suatu aplikasi yang mampu mengkonversi not balok ke not angka berdasarkan teori musik yang telah dimodifikasi sehingga konsep tidak bertentangan namun tetap sesuai dengan teori musik umum.

¹ Magister Teknologi Informasi, Universitas Gadjah Mada, Yogyakarta

² Magister Teknologi Informasi, Universitas Gadjah Mada, Yogyakarta

³ Magister Teknologi Informasi, Universitas Gadjah Mada, Yogyakarta

Proses konversi not balok ke not angka mengambil masukan dari partitur yang telah disimpan dalam bentuk file MusicXML. File MusicXML menampung susunan notasi yang bersumber dari pengetikan manual menggunakan software semacam Finale, Sibelius, dan lain sebagainya dan bisa juga berasal dari hasil scan dari partitur tercetak.

Dengan mengkhhususkan pembahasan pada partitur paduan suara, program ini mampu mengkonversi partitur paduan suara yang berisi dua suara yang berbeda atau lebih. Hasil tersebut akan disimpan dalam format file dokumen Microsoft Word (.doc) dengan tata letak yang sesuai standar penulisan not angka untuk paduan suara campur sehingga membuka kemungkinan untuk dimodifikasi lebih lanjut. Adapun hal-hal penting yang digunakan sebagai batasan masalah adalah:

- a) Sumber yang digunakan berupa representasi not balok dalam file MusicXML, dengan susunan dua suara.
- b) Komponen yang dikonversi berupa komponen-komponen pokok, yaitu judul, pengarang, nada dasar, tanda birama, tanda tempo, notasi dan lirik. Di luar komponen tersebut tidak dikonversi.
- c) Tangga nada yang digunakan adalah tangga nada mayor diatonis. Tangga nada di luar itu tidak digunakan.
- d) Harga notasi paling kecil adalah 1/16.
- e) Tujuan penelitian adalah menyediakan aplikasi konversi not balok ke not angka yang handal dan menyajikan satu metode konversi not yang cepat dan akurat.

2. LANDASAN TEORI

MusicXML sebagai kapasitasnya untuk merepresentasikan notasi, mampu berperan sebagai media perantara yang penting dan bermanfaat (Cunningham, 2005). Notasi yang disimpan tidak harus berasal dari sumber not balok konvensional, oleh karena yang disimpan adalah kode-kode notasi huruf, yang mana tidak tergantung tipe maupun format penulisan not inputnya. Hal ini sejalan dengan pendapat Edwards yang mengemukakan tata penulisan not alternatifnya. Berdasarkan referensi di atas, penelitian ini kemudian dikembangkan lebih lanjut.

Tiga landasan teori utama yang digunakan adalah teori musik tentang not balok dan not angka, pemahaman struktur file MusicXML serta pengetahuan mengenai struktur data array. Notasi yang digunakan secara internasional selama ratusan tahun hingga sekarang adalah notasi balok, dengan satuan berupa gambar. Pada penulisan not balok, nada dituliskan pada garis-garis yang disebut dengan garis paranada. Garis paranada terdiri dari lima garis yang sejajar dan sama panjangnya, seperti pada Gambar 1 berikut.



Gambar 1. Letak titi nada dalam garis paranada (C4-D4-E4-F4-G4-A4-B4-C5)

Notasi adalah sistem penulisan lagu, yang satuan-satuannya disebut not (Soeharto, 1988). Not itu sendiri adalah perwujudan dari nada, yang mempunyai definisi berupa bunyi yang teratur, yang mempunyai bilangan getar (frekuensi) tertentu (Sukohardi, 1997). Tinggi rendahnya bunyi (suara) bergantung pada besar kecilnya frekuensi tersebut. Tinggi rendah dan panjang pendeknya nada direpresentasikan dengan tanda yang disebut titinada atau not (Sukohardi, 1997), yang terdiri dari tujuh titinada pokok, C-D-E-F-G-A-B. Demikian pula teori mengenai harga nada, tanda birama, tempo, nada dasar atau key signature yang sudah dipakai selama ratusan tahun digunakan di sini. Kemudian dilengkapi juga dengan dasar-dasar mengenai not angka: 1, 2, 3, 4, 5, 6 dan 7 berikut juga tanda kromatis dan oktafnya. Kruis pada not balok disajikan dalam garis miring ke kanan, seperti pada 5/, sedangkan mol disajikan dalam garis miring ke kiri seperti pada 7\l. Titik digunakan sebagai penanda tinggi rendah oktaf. Titik di atas berarti oktaf tinggi, demikian pula sebaliknya.

Untuk membantu konversi, digunakan format MusicXML yang diciptakan oleh Recordare, Inc. MusicXML ini sangat membantu karena notasi balok hasil kreasi aplikasi-aplikasi penulis not balok seperti Finale, Sibelius, dan lain sebagainya dapat diekspor menjadi satu bentuk yang bisa digunakan bersama dan diambil informasinya dengan mudah.

Dari sini akan diketahui seberapa besar kegunaan penelitian, yaitu sejauh mana aplikasi dapat mengkonversi keseluruhan lagu dengan baik dan benar, dan apakah hasil konversi dapat dimodifikasi, dibaca dan dinyanyikan dengan baik dan benar pula.

Bahan penelitian yang digunakan adalah materi lagu-lagu yang menggunakan not balok sebagai media penulisannya. Bahan dapat berupa cetakan maupun *soft copy* yang berupa dokumen aplikasi editor not balok seperti Finale, Sibelius, Encore, dan lain sebagainya yang disimpan menggunakan metode penyimpanan MusicXML. Penelitian yang akan dilakukan menggunakan sebuah komputer dengan spesifikasi yang cukup untuk mengembangkan aplikasi berbasis Windows. Bahasa pemrograman yang digunakan adalah Delphi dengan kompilator Borland Delphi 7.

Urutan pengkonversian dikondisikan dengan urutan penekanan tombol sebagai berikut: Buka *File* > Proses dua suara /empat suara > Simpan Word dan ditutup dengan tombol Exit.

Hasil konversi diekspor dalam bentuk *file* Microsoft Word. Untuk menuliskan not angka dibutuhkan satu *font* khusus, yaitu *font* NADA-ARIAL yang dirancang sendiri

menggunakan *font* editor. Keterangan mengenai aturan penulisan not angka menggunakan *font* NADA-ARIAL bisa dilihat pada Tabel 1.

Tabel 1.

Aturan penulisan not angka menggunakan *font* NADA-ARIAL

Penekanan tombol	Fungsi	Contoh
12345670 .	Menuliskan notasi	1 2 3 4 5 6 7 0 .
	Menuliskan garis birama	3 5 . 3 1 2 . .
8	Membubuhkan titik bawah	58 58 13 2
9	Membubuhkan titik atas	1 3 5 19 39 19
-	Menurunkan setengah nada	5-
+	Menaikkan setengah nada	2+
=	Garis birama panjang	1 = 2 3 2 =
]	Garis 1/8	1]]2] 3 3 4]]5]
[Garis 1/16	1[] []2[] []3] 3]]4[] []5[]
\$	Mengawali <i>legato/tie</i>	4\$. . .
%	Garis <i>legato</i>	5\$ %.% %.% %4*
*	Mengakhiri <i>legato/tie</i>	. . 4*

Berdasarkan *font* tersebut kemudian dibuatlah ArrayNA yang berisi kode not angka dari oktaf terendah hingga tertinggi. Selain itu juga dibuat ArrayNDsr yang berisi dua belas nada dasar yang mungkin.

Proses yang terjadi saat membuka *file* adalah *file* dimuat ke memori dalam bentuk tree dan dipersiapkan untuk diparsing pada proses berikutnya. Kemudian ketika tombol Proses ditekan, informasi-informasi seperti judul lagu, nada dasar, tipe birama, tanda tempo, notasi tiap birama dan tiap jenis suara serta liriknya diambil, dikonversi sesuai aturan dan teori musik, kemudian ditampilkan pada obyek penampil yang tersedia untuk sementara. Berikut ini adalah bentuk dari struktur data *file* MusicXML setelah diparsing:

```
XMLDocument1.DocumentElement.ChildNodes.Nodes[node1]
    .ChildNodes.Nodes[node2]
    .ChildNodes.Nodes[node3]
    .
    .
    .ChildValues[node sasaran]
```

Setelah diproses dan tombol Simpan Word ditekan, maka program membuat obyek OLE Microsoft Word, mengatur atribut-atribut tampilan agar nyaman dilihat, kemudian mengeksport hasil konversi ke *file* yang dituju.

Untuk mengkonversi notasi balok digunakan acuan yang dijelaskan pada Tabel 2.

Tabel 2.
Acuan konversi not balok

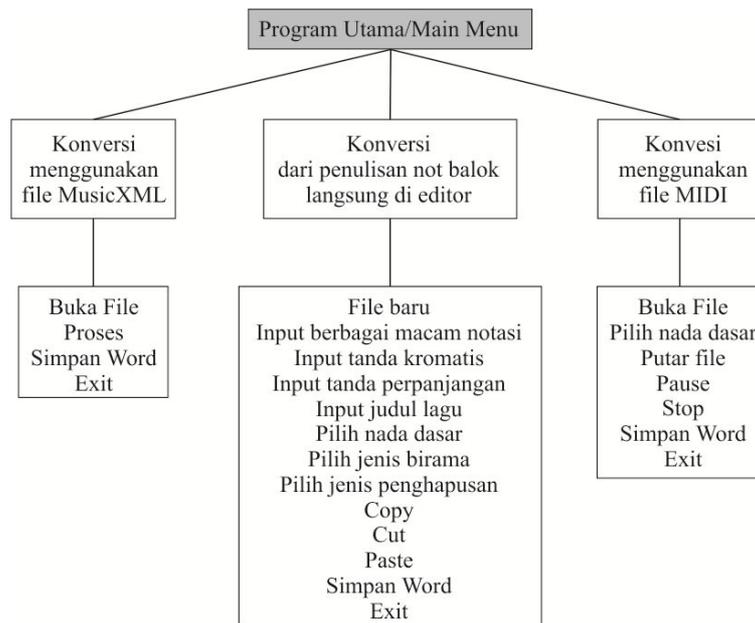
Solmisasi →		do	di	re	ri	mi	fa	fi	sol	sel	la	sa	si
Kromatis	Fifths	1	1/	2	2/	3	4	4/	5	5/	6	7\	7
0	0	C	C#/Db	D	D#/Eb	E	F	F#/Gb	G	G#/Ab	A	A#/Bb	B
1 krus	1	G	G#/Ab	A	A#/Bb	B	C	C#/Db	D	D#/Eb	E	E#/F	F#/Gb
2 krus	2	D	D#/Eb	E	E#/F	F#/Gb	G	G#/Ab	A	A#/Bb	B	B#/C	C#/Db

Untuk tanda kromatis 3 dan selanjutnya didapat dengan menjadikan nada kelima tangga nada sebelumnya sebagai nada dasar, dan menaikkan nada ketujuh setengah laras.

Dalam menggunakan metode parsing XML seperti ini, ada beberapa kesulitan yaitu ketika node yang dimaksud ternyata tidak ada, sangat beresiko program menjadi *error*. Untuk itu bagi pengembangan selanjutnya diperlukan *error trapping* yang teliti sehingga bisa menekan kesalahan yang timbul.

3. HASIL PENELITIAN DAN PEMBAHASAN

Struktur program konversi ini secara keseluruhan adalah seperti yang digambarkan pada Gambar 2 berikut.



Gambar 2. Struktur Program

Oleh karena aplikasi dikembangkan menggunakan bahasa pemrograman Delphi, struktur program di atas dibagi-bagi menjadi bagian yang disebut *unit*. Di dalam *unit* terdapat serangkaian prosedur dan fungsi untuk mengerjakan proses tertentu. Untuk kasus ini, bagian Main Menu dinamai sebagai Unit **main**, bagian konversi menggunakan

menggunakan *file* MusicXML sebagai Unit **konversiXML**, bagian konversi dari penulisan not balok langsung di editor sebagai Unit **anak**, dan bagian konversi menggunakan *file* MIDI sebagai Unit **konversiMIDI**.

Konversi menggunakan *file* MusicXML dipilih untuk dikembangkan lebih lanjut karena format *file* MusicXML adalah format yang paling lengkap dan detail dalam merepresentasikan not balok. Alasan yang mendasari mengapa jenis konversi yang kedua dan ketiga tidak/belum dikembangkan lebih lanjut adalah karena keterbatasan dalam merancang design *interface* yang menyerupai editor not balok yang sudah beredar di pasaran. Sedangkan *file* MIDI sendiri dalam penyimpanan informasinya lebih kompleks daripada *file* MusicXML.

Sebelum masuk ke prosedur-prosedur dalam unit konversiXML, terlebih dulu didefinisikan variabel-variabel dan array yang akan digunakan. Berikut ini adalah ArrayNA yang jika komponennya ditulis menggunakan *font* NADA-ARIAL akan memunculkan not angka dengan format penulisan yang standar.

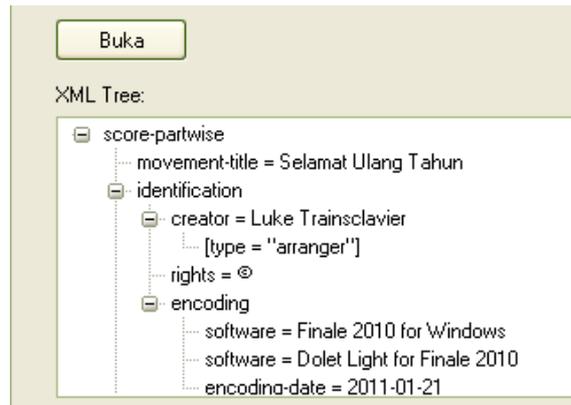
```
ArrayNA : array[1..60] of string =
('18 8', '1+8 8', '28 8', '2+8 8', '38 8', '48 8', '4+8 8',
'58 8', '5+8 8', '68 8', '7-8 8', '78 8',
'18', '1+8', '28', '2+8', '38', '48', '4+8', '58', '5+8', '68',
'7-8', '78', '1', '1+', '2', '2+', '3', '4', '4+', '5', '5+', '6',
'7-', '7', '19', '1+9', '29', '2+9', '39', '49', '4+9', '59',
'5+9', '69', '7-9', '79', '19 9', '1+9 9', '29 9', '2+9 9', '39 9',
'49 9', '4+9 9', '59 9', '5+9 9', '69 9', '7-9 9', '79 9');
```

ArrayNDsr adalah array yang berisi nada dasar, berfungsi untuk menuliskan nada dasar sesuai dengan pointer yang ditunjukkan oleh nilai *node fifth* dalam MusicXML.

```
ArrayNDsr : array[1..12] of string =
('C', 'C#/Db', 'D', 'D#/Eb', 'E', 'F', 'F#/Gb', 'G', 'G#/Ab', 'A', 'A#/Bb',
', 'B');
```

Kedua array itulah yang memegang peranan penting dalam konversi berikutnya.

Selanjutnya adalah membuat prosedur untuk membuka *file* MusicXML dengan langkah-langkah membuka folder yang aktif, menjalankan *progressbar*, *load from file*, muat ke *tree* dengan prosedur DOMToTree, tampilkan *tree* dengan semua cabang terbuka, aktifkan button Proses 2 Suara maupun 4 Suara. Tak lupa ditambahkan *error trap* untuk mencegah *file* non MusicXML yang dibuka. Prosedur DOMToTree sendiri adalah prosedur yang menelusuri *node* demi *node* untuk mengambil nama *node*, atribut beserta nilainya untuk dimuat ke dalam bentuk *tree*. Hasil dari langkah-langkah di atas adalah *file* MusicXML dibuka dan di-parse ke dalam *tree* yang ditampilkan di bagian kiri atas layar seperti pada Gambar 3 berikut:



Gambar 3. TreeXML

Kemudian ketika tombol Proses ditekan maka langkah yang dilakukan adalah mengambil informasi judul pada:

```
XMLDocument1.DocumentElement.ChildValues['movement-title'],
```

pengarang pada:

```
XMLDocument1.DocumentElement.ChildNodes.Nodes['identification'].ChildValues['creator'],
```

nada dasar pada:

```
XMLDocument1.DocumentElement.ChildNodes.Nodes[4].ChildNodes.Nodes['measure'].ChildNodes.Nodes['attributes'].ChildNodes.Nodes['key'].ChildValues['fifths'].
```

Di sinilah pointer *fifth* pada MusicXML merujuk pada nada dasar yang digunakan. Nilai *fifth*=0 berarti C, 1 berarti G, 2 berarti D, 3 berarti A, 4 berarti E, 5 berarti B, 6 berarti Fis, -6 berarti Ges, -5 berarti Des, -4 berarti As, -3 berarti Es, -2 berarti Bes, dan -1 berarti F. Nilai *step* di atas menunjukkan jumlah pergeseran pointer untuk ArrayNA, agar do dalam not angka bisa berada tepat pada nada dasar yang sudah ditentukan, serta not-not lain seperti re, mi, fa, sol, la, si juga menyesuaikan.

Yang tidak kalah pentingnya adalah informasi tipe birama:

```
XMLDocument1.DocumentElement.ChildNodes.Nodes[4].ChildNodes.Nodes['measure'].ChildNodes.Nodes['attributes'].ChildNodes.Nodes['time'].ChildValues['beats'] dan .ChildValues['beat-type'].
```

Tanda tempo:

```
XMLDocument1.DocumentElement.ChildNodes.Nodes[4].ChildNodes.Nodes['measure'].ChildNodes.Nodes['direction'].ChildNodes.Nodes['direction-type'].ChildValues['words']
```

dan

```
XMLDocument1.DocumentElement.ChildNodes.Nodes[4].ChildNodes.Nodes  
[ 'measure' ].ChildNodes.Nodes[ 'direction' ].ChildNodes.Nodes[ 'sound  
' ].Attributes[ 'tempo' ].
```

Langkah berikutnya adalah mengecek apakah not yang dijumpai merupakan not *rest*/tanda diam. Dengan menggunakan logika *if* ditentukan harga not *rest* yang cocok dengan sumbernya sesuai teori musik.

Kemudian akan dicek apakah not berupa not C, D, E, F, G, A atau B. Pada tiap not, juga dicek tanda kromatisnya, bila dijumpai '*alter*' = 1 berarti mendapat krus/naik setengah nada dan di ArrayNA digeser 1 step ke kanan. Sebaliknya, jika dijumpai '*alter*' = -1 berarti mendapat mol/turun setengah nada dan di ArrayNA digeser 1 step ke kiri. Kemudian dicek tinggi rendah oktafnya. Jika '*octave*' di angka 3, maka diambil ArrayNA di oktaf rendah. Jika '*octave*' di angka 4, maka diambil ArrayNA di oktaf netral. Jika '*octave*' di angka 5 maka ArrayNA di oktaf tinggi. Baris perintah

```
RichEdit2.SelText := ArrayNA[...+step]
```

adalah perintah untuk merujuk pada indeks array dalam ArrayNA. Angka indeks ini berbeda-beda tergantung nada yang ditemukan. Setelah mengambil informasi nadanya, langkah selanjutnya adalah mengambil informasi harga nadanya, menandai tipe triol dan legato.

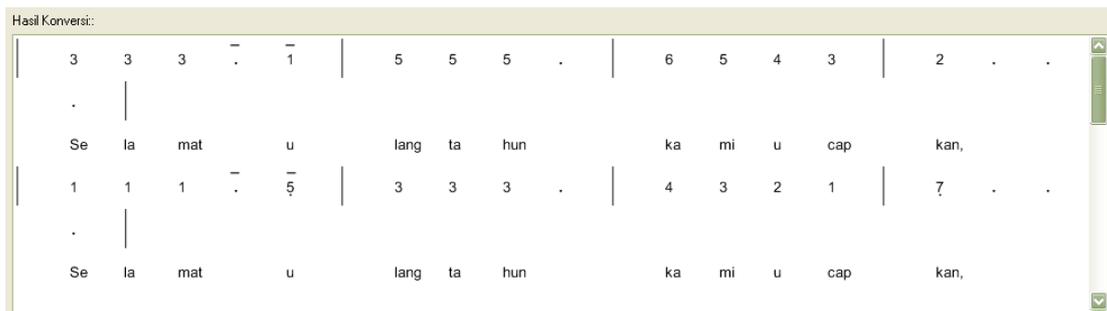
Setelah notasi dikonversi per empat birama, langkah selanjutnya adalah mengambil informasi lirik yang terdapat pada

```
RichEdit2.SelText:=XMLDocument1.DocumentElement.ChildNodes.Nodes[  
4].ChildNodes.Nodes[m].ChildNodes.Nodes[x].ChildNodes.Nodes[ 'lyric'  
' ].ChildValues[ 'text' ].
```

Dalam mengambil lirik juga diperhatikan harga notasi dan pemisahan birama sehingga kerapian partitur hasil konversi dapat terjaga.

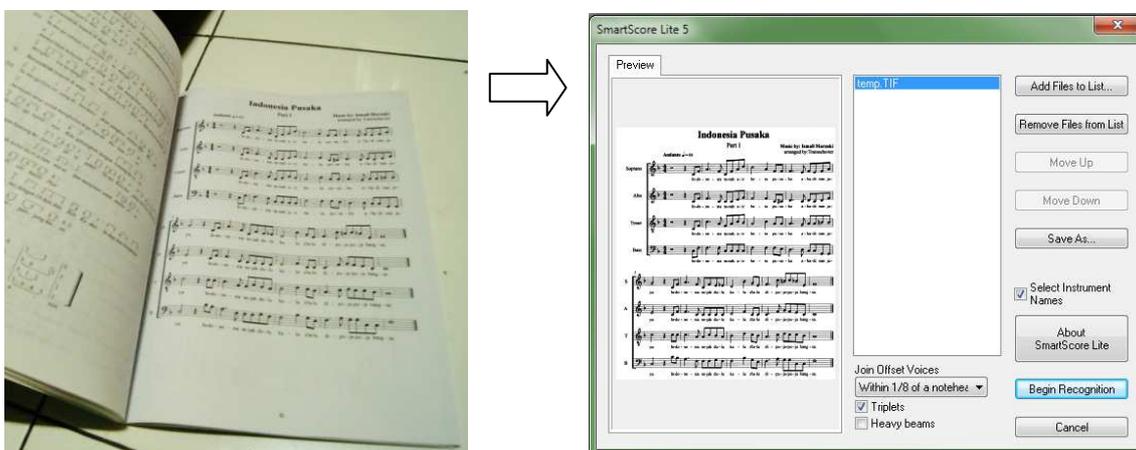
Setiap pengambilan informasi di atas langsung ditampilkan di RichEdit2 di layar bagian tengah, seperti yang ditampilkan pada Gambar 4.

Usai dikonversi, *user* diarahkan untuk menyimpannya dalam bentuk *file* Microsoft Word. Langkah-langkahnya adalah dengan membuat OLEObject Word.Application, mengatur atribut-atribut tampilan seperti nama *font*, ukuran, lebar tabulasi, dan spasi antar baris. Dalam hal ini pengaturan atribut untuk judul dibedakan dengan atribut untuk notasi. Hasil dari tampilan di RichEdit2 diekspor menggunakan cara berikut: **WordApplication.Selection.TypeText(Richedit2.Text)**. Setelah berhasil kemudian langsung membuka dialog Save As untuk menyimpan *file* hasil konversi.



Gambar 4. RichEdit2 sebagai object untuk meletakkan hasil konversi.

Berikut ini contoh lagu “Indonesia Pusaka” ciptaan Ismail Marzuki. Partitur lagu ini merupakan arsip pribadi dan sudah tidak memiliki *file softcopy*-nya lagi sehingga perlu dipindai menggunakan *scanner*. Proses pemindaian dalam Finale dibantu oleh program SmartScore versi 5.7.0 buatan Musitek Corporation. Langkah-langkahnya adalah kita memilih menu Create New Music > Scanning > Preview partitur yang akan dipindai > Final Scan > Begin Recognition > Pemilihan suara untuk playback. Pada Gambar 5 berikut ditampilkan langkah-langkahnya.



Gambar 5. Proses pemindaian partitur

Setelah berhasil dibuka di Finale, langkah selanjutnya adalah mengekspor ke *file MusicXML* seperti langkah-langkah yang telah dijelaskan sebelumnya. Kemudian dari *MusicXML* dibuka dengan program konversi, kemudian dilakukan proses konversi dan hasilnya disimpan dalam *file Microsoft Word* seperti pada Gambar 5.

Indonesia Pusaka
Ismail Marzuki

Do=F, 4/4
Andante (MM=60)

0 0 0 $\bar{5}$ $\bar{1}$	3 . 1 $\bar{5}$ 1 3 $\bar{6}$	5 . 3 $\bar{1}$ 1	1 . $\bar{7}$ 1 $\bar{7}$ 1 3
	In do	ne	sia ta naha ir
0 0 0 $\bar{5}$ $\bar{1}$	1 . $\bar{5}$ $\bar{5}$ $\bar{5}$ $\bar{7}$ 2	1 . 1 $\bar{5}$ $\bar{7}$	6 . $\bar{5}$ $\bar{6}$ $\bar{5}$ $\bar{6}$ 1
	In do	ne	sia ta naha ir
2 . 0 $\bar{5}$ $\bar{1}$	3 . 1 $\bar{5}$ 1 $\bar{7}$ $\bar{8}$	6 . 4 2 $\bar{7}$	1 . $\bar{5}$ $\bar{7}$ $\bar{5}$ 4 $\bar{7}$
ya.	In do	ne	sia se jak du lu
			ka la s'la lu di pu ja pu ja bang

Gambar 6. Hasil konversi partitur lagu “Indonesia Pusaka”

Dari hasil penelitian, akan diujikan sesuai metode pengujian rekayasa perangkat lunak, yaitu *Black Box Testing* dan *White Box Testing*. Untuk *Black Box Testing*, diberikan input yang tidak sesuai dengan persyaratan. Hasil yang didapat adalah muncul banyak pesan kesalahan. Kemudian untuk *White Box Testing* digunakan materi input not balok sesuai persyaratan, yaitu lagu Indonesia Pusaka, karangan Ismail Marzuki, dengan nada dasar Do=F, 4/4, tempo Andante (MM=60), jumlah suara 4 dengan masing-masing terdapat lirik, jumlah birama 9. Dari pengujian didapat bahwa struktur program antar prosedur berupa linier namun pada prosedur Proses konversi, dijumpai percabangan dan perulangan dalam jumlah yang cukup banyak. Hal ini mengingatkan pentingnya mengakomodasi berbagai kemungkinan penulisan notasi yang sering dijumpai.

Berdasarkan uji coba di atas, proses parsing *file* MusicXML memakan waktu relatif lebih lama daripada proses konversi itu sendiri. Hal ini terlihat pada saat membuka *file* dan parsing memakan waktu sepanjang sepuluh hingga dua puluh detik. Sedangkan proses konversi hanya memakan waktu satu-dua detik saja pada PC dengan spesifikasi Intel Core2Duo E6300 1.86 GHz dan memori 3 GB. Sistem dengan prosesor yang lebih cepat dan memori yang lebih lapang mampu memproses dengan lebih singkat pula.

4. KESIMPULAN DAN SARAN

Dari hasil pengujian yang didapat, berikut ini adalah kelebihan-kelebihan yang dari program:

- a) mengkonversi not balok dengan akurat sesuai dengan batasan penelitian, dengan perantara *file MusicXML*;
- b) mengkonversi not balok dengan jumlah suara lebih dari satu;
- c) menyertakan informasi-informasi tambahan seperti judul, pengarang, nada dasar, birama dan tempo dengan akurat;
- d) menyertakan lirik dalam proses konversi;
- e) menyimpan hasil konversi dalam format yang umum dan mudah diedit oleh *user*

Beberapa kelemahan yang didapat yaitu:

- a) masih perlu dilengkapi dengan *error-trapping* yang mampu menutup celah-celah program menjadi *crash* karena dianggap salah melakukan *parsing* partitur not balok yang di luar batasan penelitian;
- b) karena keterbatasan desain, program belum mampu menyatukan kelompok not 1/8 dan 1/16, memasukkan tanda dinamika, dan meletakkan lirik secara akurat penuh;
- c) masih perlu menambahkan kemampuan untuk mengenali berapa jumlah suara yang akan dikonversi dengan tujuan meminimalisasi munculnya kesalahan dalam proses konversi.

Daftar Pustaka

- Cantu, M. (2003). *Mastering Delphi 7*. Alameda: Sybex.
- Cunningham, S. 2005. *Suitability Of MusicXML As A Format For Computer Music Notation & Interchange*. University of Wales: North Wales.
- Joyopuspito, S. (2004). *Ilmu Harmoni Musik*. Jakarta: Bina Musik Remaja.
- Koniaris, *Understanding Musical Notes and their Notation*. Diakses pada 20 Februari 2010 dari World Wide Web: <http://www.koniaris.com/music/notes/>
- Prier, K.E. (1979). *Ilmu Harmoni*. Pusat Musik Liturgi Yogyakarta. Edisi XIV 2006
- Purwacandra, P.P. (2008). *Asyik Berkreasi dengan MIDI*. Yogyakarta: Penerbit Andi.
- Recordare. (2003). *MusicXML Definition*. Diakses pada 4 April 2010 dari World Wide Web: <http://www.makemusic.com/musicxml/publications/recordare>
- Révész, G. (2001). *Introduction to the Psychology of Music*. New York: Dover Publications.
- Soeharto, M. (1988). *Belajar Notasi Balok*. Jakarta: PT. Gramedia
- St Laurent, S. (2000). *XML Elements of Style*, Berkshire: Hill Publishers.
- Sukohardi, Al. (1997). *Teori Musik Umum*. Yogyakarta: Pusat Musik Liturgi.
- The Music Notation Project. *Evaluative Research on Alternative Music Notations*. Diakses pada 15 Juni 2010 dari World Wide Web <http://musicnotation.org/musicnotations/research.html>