

# Penyembunyian Pesan dalam File MIDI

Stefanus 13519101

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail : 13519101@std.stei.itb.ac.id

**Abstrak**—MIDI (*Musical Instrument Digital Interface*) adalah tipe data yang dijadikan acuan dasar perangkat keras dan perangkat lunak untuk saling bertukar data antara perangkat musik dengan komputer. MIDI berbeda dengan musik dengan tipe file MP3 ataupun WAV yang memiliki isi data berupa representasi gelombang suara. MIDI berisikan representasi notasi musik termasuk pitch, tempo, dan dinamika yang dapat dimainkan sembarang perangkat elektronik yang memiliki synthesizer untuk memberikan karakter alat musik.

**Kata Kunci**—MIDI; Steganografi; Algoritma

## I. PENDAHULUAN

Kata orang, musik adalah bahasa universal karena dapat diterima dan dimengerti oleh semua kalangan. Katanya juga, musik dapat menyampaikan pesan kepada seluruh pendengar dan penikmat musik itu sendiri. Musik memegang peranan yang penting di berbagai aspek kehidupan manusia. Baik sebagai alat peneduh jiwa, pelipur lara, sampai pemenuhan kebutuhan hasrat manusia akan seni dan berkreasi. Musik juga acap kali digunakan sebagai alat berkomunikasi antara pencipta musik, pemain musik, dengan penikmat dan pendengar musik. Hal ini dapat terjadi karena dengan alunan nada, musik mengungkapkan pikiran, isi hati, dan perasaan manusia dalam bentuk gelombang-gelombang nada harmonis yang langsung merasuk kalbu.

Tanpa disadari, musik sudah tidak dapat dipisahkan dari kehidupan manusia itu sendiri. Bayangkan iklan produk yang sering muncul di video YouTube, kehilangan musiknya. Pasti akan menjadi gambar bergerak yang membosankan nan menjengkelkan. Namun, apa jadinya bila musik dijadikan alat komunikasi yang secara nyata menyembunyikan pesan di dalamnya, di mana bukanlah pencipta, pemain, maupun penikmat musik yang dapat mengerti pesannya, melainkan hanya orang yang tahu bahwa ada pesan di dalamnya-lah yang dapat mengetahui dan mengerti isi pesan tersebut.

Steganografi adalah teknik penyembunyian data rahasia ke dalam media data lain sehingga keberadaan data rahasia tidak diketahui atau disadari oleh orang lain. Steganografi terhadap file MIDI merupakan konsep yang muncul di benak penulis ketika membayangkan musik sebagai penyampai pesan, namun bukan dengan alunan nada, melainkan dengan alunan *binary* dan karakter yang tersembunyi, tidak disadari, yang ada di dalam notasi-notasi musik yang ada.

## II. LANDASAN TEORI

### A. Steganografi

Steganografi merupakan teknik penyembunyian pesan rahasia ke dalam pesan lainnya sebagai media sedemikian rupa sehingga keberadaan dari pesan rahasia tersebut tidak disadari atau diketahui keberadaannya oleh orang lain. Steganografi biasa terdiri dari dua properti, yakni pesan sebagai wadah penampung, dan pesan rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital baik sebagai wadah penampung maupun data rahasia, misalnya citra, suara (audio), teks, dan video. Steganografi konvensional biasanya merupakan usaha untuk merahasiakan komunikasi dengan cara menyembunyikan pesan ataupun menyamarkan pesan. Sehingga prinsip dasar dalam steganografi adalah lebih dikonsentrasikan pada kerahasiaan pesannya bukan pada data mediana.

Ada beberapa kriteria yang harus diperhatikan dalam melakukan proses steganografi, yaitu :

- *Imperceptibility*, keberadaan data rahasia tidak dapat dipersepsi oleh indera manusia. Citra stego harus tidak dapat dibedakan dengan citra cover.
- *Fidelity*, kualitas data penampung tidak banyak berubah setelah dilakukan penyisipan.
- *Recovery*, data rahasia yang disembunyikan harus dapat diungkap kembali
- *Capacity*, berhubungan dengan jumlah informasi yang dapat disisipkan ke dalam data penampung.
- *Robustness*, data yang disembunyikan harus tahan dari berbagai operasi manipulasi yang dilakukan pada data penampung.
- *Undetectability*, kemampuan untuk menghindari deteksi baik oleh indera manusia maupun analisis statistik.

Dalam melakukan proses steganografi, ada beberapa faktor yang saling berkompetensi satu sama lain (*trade-off*), artinya saat salah satu faktor ditingkatkan maka kemungkinan faktor lain akan mengalami penurunan.

### B. MIDI

MIDI adalah sebuah standar *hardware* dan *software* internasional untuk saling bertukar data (seperti kode musik dan *MIDI Event*) diantara perangkat musik elektronik dan komputer dari merek yang berbeda, yakni merupakan bahasa

yang digunakan instrumen musik elektrik, pengendali, komputer, dan peranti sejenis untuk berkomunikasi. MIDI juga mengandung spesifikasi peranti keras yang memungkinkan alat-alat tersebut terhubung.

MIDI menangkap *event* notasi dan perubahan atribut dan aksentuasi, mengkodekannya menjadi pesan digital, dan mengirimkan kode tersebut sebagai pesan ke peranti lain untuk mengatur suara yang dihasilkan beserta parameternya. Data jenis ini dapat direkam dengan *sequencer*, seperti Cubase, Fruity Loops, atau Cakewalk. MIDI membawa pesan event musikal yang terdiri dari notasi, *pitch* and *velocity* (tekanan), sinyal pengendali seperti volume, *vibrato*, *audio panning* dan *cues*, juga *clock signals* yang diatur mensinkronkan tempo antara berbagai peranti musik. Satu file MIDI dapat memuat enam belas channel informasi musik yang masing-masing dapat di arahkan ke peranti yang berbeda. Dari enam belas channel tersebut, channel sepuluh khusus dipergunakan untuk perkusi.

Di era tahun 1970-an perkembangan elektrik piano atau yang biasa disebut dengan “*keyboard*” berkembang dengan pesat. Elektrik piano merupakan sebuah instrumen piano yang mempunyai bahasa pemrograman yang tersimpan didalam microchip sehingga dapat diubah-ubah suaranya sesuai dengan kebutuhan kita. Dimana pada saat itu setiap produsen instrumen musik mempunyai bahasa pemrograman sendiri-sendiri.

MIDI *event* terdiri dari beberapa tipe, yakni :

No	Tipe MIDI Event	Fungsi
1	Note Off Event	Memberikan informasi mengenai not yang berhenti ditekan
2	Note On Event	Memberikan informasi mengenai not yang sedang ditekan
3	Note Aftertouch Event	Memberikan informasi mengenai tekanan/dinamika dari not yang sedang ditekan
4	Controller Event	Memberikan informasi perubahan dari satu <i>channel</i> MIDI
5	Program Change Event	Memberikan informasi mengenai perubahan dari instrumen MIDI
6	Channel Aftertouch Event	Memberikan informasi mengenai tekanan/dinamika dari <i>channel</i>
7	Pitch Bend Event	Memberikan informasi mengenai perubahan <i>pitch</i> dari <i>channel</i> .

### III. PEMBAHASAN

Dalam pengerjaan makalah ini, penulis akan menggunakan program *python* dalam proses steganografi dan ekstraksi pesannya, serta menggunakan aplikasi *MuseScore 3.0* dalam visualisasi dari MIDI itu sendiri sedemikian rupa dapat divisualisasikan dalam bentuk partitur.

Ide yang muncul di benak penulis adalah membuat banyak notasi di bagian awal MIDI dengan *velocity* 0 sedemikian rupa sehingga tidak mengubah kualitas dari hasil *export* MIDI menjadi *file* MP3 ataupun WAV.

Disini penulis akan menggunakan aransemen lagu “Balada Seorang Biduan” karya Bimbo dari inventaris PSM-ITB. Pertama penulis akan membuat MIDI dari aransemen lagu “Balada Seorang Biduan”, setelah itu membuat program *python* untuk menyisipkan notasi-notasi dengan *velocity* 0, kemudian mengisi data instrumen dari notasi-notasi ini dengan data dari *plaintext* yang ada.

#### A. Membuat MIDI dari partitur

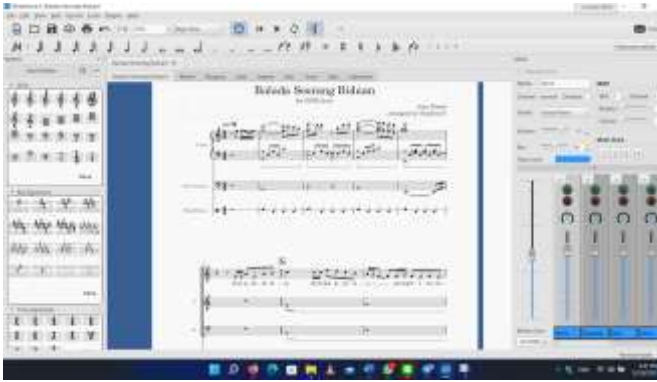
Hal yang pertama diperlukan dalam steganografi adalah pesan wadah. Dalam makalah ini wadah yang digunakan adalah file MIDI. Disini berdasarkan partitur lagu “Balada Seorang Biduan”, penulis membuat file MIDI menggunakan aplikasi *MuseScore*. Hal pertama yang harus dimasukkan adalah memasukkan notasi-notasi yang tertera di dalam partitur ke dalam aplikasi *MuseScore*. Berikut adalah gambaran partitur dari lagu “Balada Seorang Biduan”.



Gambar 3.1 Partitur Balada Seorang Biduan

Sumber : PSM-ITB

Setelah itu, masukkan satu per satu notasi yang ada ke dalam program *museScore* sedemikian rupa sehingga tampilan *MuseScore* menjadi seperti berikut :



Gambar 3.2 MuseScore Balada Seorang Biduan

Sumber : Pribadi

Setelah itu, menggunakan fitur *museScore*, buat file MIDI berdasarkan partitur yang sudah ditulis melalui “File > Export > Export To > MIDI File.”

### B. Mengecek Keberadaan track dari file MIDI

Dalam mengecek keberadaan *track* dari file MIDI, digunakan *library python* yaitu MIDI.

```
tracks = midi.read_midifile(midi_file)
if len(tracks) == 0:
    tracks.append(midi.Track)
```

Demikian rupa sehingga terjamin setidaknya ada 1 *track* yang berada di dalam file MIDI.

### C. Membuat flag mulai steganografi

Disini, penulis akan menyisipkan flag berupa NoteOffEvent dengan tick 0 ms dan nada sebagai flag berupa G3 (Middle G). Hal ini guna mempermudah ekstraksi pesan dan penghapusan pesan dari file MIDI.

```
off = midi.NoteOffEvent(tick=0,
pitch=midi.G_3)
tracks[0].insert(0, off)
```

Sedemikian rupa sehingga dapat diketahui bila ada not off sepanjang 0 ms dengan nada G3, not setelahnya adalah not yang mengandung data berupa *plaintext*.

### D. Menyisipkan pesan

Setelah memberikan *flag* penentu mulainya steganografi, selanjutnya adalah menyisipkan pesan. Hal ini dapat dilakukan dengan cara untuk tiap karakter di dalam *plaintext*, di-*append* satu persatu *ord*-nya sedemikian rupa sehingga *track* saat ini berisikan *flag* mulai pesan serta pesannya itu sendiri.

```
for character in message:
    change_program =
midi.ProgramChangeEvent(tick=0,
data=[ord(character)])
tracks[0].insert(0,
change_program)
```

### E. Membuat flag akhir steganografi

Disini, penulis akan menyisipkan flag berupa NoteOnEvent dengan tick 0 ms dan nada sebagai flag berupa G3 (Middle G). Hal ini guna mempermudah ekstraksi pesan dan penghapusan pesan dari file MIDI.

```
on = midi.NoteOnEvent(tick=0,
pitch=midi.G_3)
tracks[0].insert(0, on)
```

Sedemikian rupa sehingga dapat diketahui bila ada not on sepanjang 0 ms dengan nada G3, not setelahnya bukan lagi not yang mengandung data berupa *plaintext*.

### F. Menyimpan track yang sudah di edit ke dalam MIDI

Setelah selesai memanipulasi *track*, perlu disimpan kembali ke dalam MIDI dengan perintah :

```
midi.write_midifile(midi_file,
tracks)
```

Sedemikian rupa sehingga file MIDI (.mid) saat ini sudah berisikan *tracks* yang dimanipulasi.

### G. Mencari Flag Akhir Pesan

Karena penyisipannya dari depan ke belakang, maka dalam ekstraksinya hasil event-nya akan terbalik, sedemikian rupa sehingga hal pertama yang harus dicari adalah flag on. Hal ini dilakukan dengan cara mengiterasi tiap event yang ada untuk mencari flag on.

```
for event in tracks[0]:
    i = 0
    if isinstance(event,
midi.NoteOnEvent) and event.tick ==
0 and event.get_pitch() == midi.G_3:
        secrets[secret_index] =
list()
```

Sedemikian rupa sehingga tercatat di posisi mana flag on ditemukan, dan dibuat list guna mengisi *ord* dari hasil ekstraksi selanjutnya.

### H. Mengekstraksi Plaintext

Disini, penulis akan mulai mengekstraksi satu persatu *ord* dari *plaintext* yang terenkripsi di belakang *flag on*. Data *plain*

*text* ini akan disimpan ke dalam list yang sudah dibuat sebelumnya.

```
for event in tracks[0]:
    if isinstance(event,
midi.ProgramChangeEvent) and
event.tick == 0:
        secrets[secret_index].append(event.data[0])
```

Sedemikian rupa sehingga saat ini list berisikan *ord* dari *plaintext* dengan terurut terbalik. Proses ini akan dilanjutkan terus menerus hingga menemukan *flag off*.

#### I. Mencari Flag Awal Pesan

Setelah mencatat *plain text* dalam urutan terbalik, proses akan dihentikan ketika sudah menemukan *flag on* berupa MIDI NoteOffEvent.

```
for event in tracks[0]:
    if isinstance(event,
midi.NoteOffEvent) and event.tick ==
0 and event.get_pitch() == midi.G_3:
        secret_index += 1
        forget_indexes.append(i)
```

Sedemikian rupa sehingga dapat diketahui bila ada not off sepanjang 0 ms dengan nada G3, event selanjutnya sudah bukan bagian dari *plain text* saat ini.

#### J. Membalik dan Mengembalikan Plaintext

Karena saat ini *plaintext* dalam bentuk *ord* dan berurutan terbalik, perlu dikembalikan dengan cara mengiterasi satu persatu sambil mengembalikan dalam bentuk *char*..

```
for i in secrets:
    secret = ''
    for char in
reversed(secrets[i]):
        secret = secret + chr(char)
    print (secret)
```

Sedemikian sehingga saat ini variabel *secret* sudah berisikan *plaintext* yang sesungguhnya.

#### IV. KESIMPULAN DAN DEMO

Berdasarkan kepada seluruh fungsi yang dibuat, maka akan menjadi sebuah *source code* utuh yang dapat dijalankan dalam bahasa pemrograman *python*. Berikut akan ditampilkan *source code* utuh yang mampu menyembunyikan pesan ke dalam file MIDI.

```
from docopt import docopt
import midi

VERSION = 1.0

def hide_message(midi_file, message):
    print ("Hide message: \"\" +
message + "\" in \"\" + midi_file)
    try:
        tracks =
midi.read_midifile(midi_file)
        if len(tracks) == 0:
            tracks.append(midi.Track)
            off =
midi.NoteOffEvent(tick=0,
pitch=midi.G_3)
            tracks[0].insert(0, off)
            for character in message:
                change_program =
midi.ProgramChangeEvent(tick=0,
data=[ord(character)])
                tracks[0].insert(0,
change_program)
                on =
midi.NoteOnEvent(tick=0,
pitch=midi.G_3)
                tracks[0].insert(0, on)
            midi.write_midifile(midi_file,
tracks)

    except IOError:
        print ("Error reading the
input file")
```

```

def reveal_message(midi_file, forget):
    print ("Reveal message in " +
midi_file)
    tracks =
midi.read_midifile(midi_file)
    secrets = {}
    secret_index = 0
    forget_indexes = list()
    for event in tracks[0]:
        i = 0
        if isinstance(event,
midi.NoteOnEvent) and event.tick == 0
and event.get_pitch() == midi.G_3:
            secrets[secret_index] =
list()
            forget_indexes.append(i)

            if isinstance(event,
midi.ProgramChangeEvent) and
event.tick == 0:
                secrets[secret_index].appe
nd(event.data[0])
                forget_indexes.append(i)

            if isinstance(event,
midi.NoteOffEvent) and event.tick == 0
and event.get_pitch() == midi.G_3:
                secret_index += 1
                forget_indexes.append(i)
            i += 1

    for i in secrets:
        secret = ''
        for char in
reversed(secrets[i]):
            secret = secret +
chr(char)
            print (secret)

    if forget:
        for i in forget_indexes:
            tracks[0].pop(i)
            midi.write_midifile(midi_file,
tracks)

```

```

if __name__ == '__main__':
    options = docopt(__doc__,
version=VERSION)
    arguments = docopt(__doc__)
    if arguments['--hide']:
        hide_message(arguments['--
file'], arguments['--message'][0])
    if arguments['--reveal']:
        reveal_message(arguments['--
file'], arguments['--forget'])

```

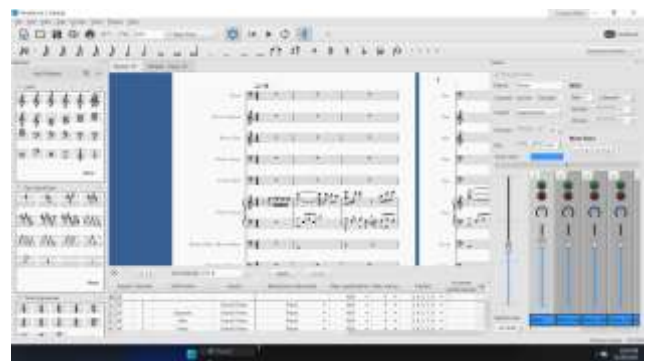
Berdasarkan *source code* ini, berikut diberikan 1 contoh permainan penyembunyian file :



Gambar 4.1 Menyembunyikan Pesan

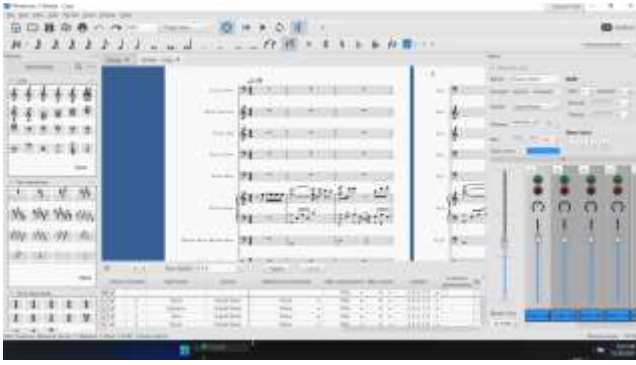
Sumber Pribadi

Disini penulis mencoba menyembunyikan pesan berbunyi “Derapakan Langkah Tatap Kedepan ITB Citra Ganesha”. Setelahnya penulis mencoba melihat partitur hasil perubahan steganografi ini.



Gambar 4.2 Partitur tanpa Steganografi

Sumber Pribadi



Gambar 4.3 Partitur dengan Steganografi

Sumber Pribadi

Terdapat perbedaan dari instrumen yang digunakan pada *track* paling atas sedemikian rupa sehingga muncul instrumen tambahan yaitu “Voice”. Namun hal ini tidak memengaruhi hasil dari MP3 karena instrumen “Voice” hanya muncul 0 ms (berarti tidak dimainkan).

Karena sudah terbukti tidak memengaruhi MIDI-nya, penulis akan mencoba mengekstrak pesan dari file MIDI tersebut.



Gambar 4.4 Ekstraksi Pesan

Sumber Pribadi

Setelah memanggil perintah ekstraksi, muncul pesan yang disembunyikan, yakni “Derapkan Langkah Tatap Kedepan ITB Citra Ganesha”.

Maka kesimpulannya adalah, dengan menambahkan instrumen dengan *velocity* 0 dan waktu 0 ms, maka kita dapat

menyembunyikan pesan ke dalam file MIDI tanpa mengubah kualitas dari file MIDI itu sendiri.

#### UCAPAN TERIMA KASIH

Terima kasih sebesar-besarnya kepada Kopi Dari Hati yang sudah rela dibisingkan dengan MIDI lagu dangdut lawas serta papan ketik mekanikal selama kurang lebih 5 jam hanya dengan 1 gelas *Affogato* dan 1 buah *Croissant*, juga kepada Dr. Ir. Rinaldi Munir, MT. yang sangat berjasa bagi diri penulis dengan materi ajar yang sangat *mindblowing* sehingga penulis dapat menyelesaikan makalah ini dengan relatif baik.

#### REFERENSI

- [1] <http://repository.unpas.ac.id/27634/5/8.BAB%20I.pdf> diakses pada 11 Desember 2021.
- [2] [http://eprints.undip.ac.id/32526/1/Anton\\_Prabowo.pdf](http://eprints.undip.ac.id/32526/1/Anton_Prabowo.pdf), diakses pada 11 Desember 2021
- [3] [https://www.mixagesoftware.com/en/midikit/help/HTML/midi\\_events.html](https://www.mixagesoftware.com/en/midikit/help/HTML/midi_events.html), diakses pada 11 Desember 2021

#### 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, 20 Desember 2021

Stefanus 1351910