AUDIO FINGERPRINT UNTUK IDENTIFIKASI FILE AUDIO

Stefanus Irwan Yuanto[1], Junius Karel T[2], Restyandito[3]

Abstrak: Identifikasi file audio secara biner kurang efektif karena adanya format penyimpanan dan cara penyimpanan file audio yang berbeda-beda. Dengan menerapkan konsep *audio fingerprint* maka sinyal audio akan diidentifikasi dengan membandingkan sebuah kode unik berukuran kecil yang mewakili isi sinyal audio tersebut sehingga perbedaan format dan cara penyimpanan tidak berpengaruh besar terhadap sebuah proses identifikasi audio.

Kata Kunci: *Audio Fingerprint, Acoustic Fingerprint, Robust Audio Identification*

1. Pendahuluan

Salah satu cara yang biasa digunakan untuk identifikasi file adalah dengan membandingkan data secara biner. Pada file audio, cara ini kurang efektif karena adanya perbedaan format, dan cara penyimpanan file audio tersebut sehingga jika identifikasi file audio dilakukan secara biner, maka dua buah sinyal audio yang terdengar sama oleh telinga namun berbeda format atau cara penyimpanannya akan dianggap berbeda.

Audio Fingerprint adalah sebuah kode unik berukuran kecil yang akan mewakili sinyal audio. Untuk membentuk sebuah *audio fingerprint* yang mewakili sebuah sinyal audio, maka isi (*content*) dari sinyal audio harus diekstrak. Isi audio ini kemudian akan dihitung energinya yang kemudian akan diubah menjadi sebuah *audio fingerprint*. Identifikasi dengan menggunakan *audio fingerprint* diharapkan mampu mengenali sinyal audio yang mengalami modifikasi baik terhadap isinya maupun format penyimpanannya.

2. Landasan Teori

Audio fingerprint[4] atau yang juga disebut *acoustic fingerprint* adalah kode unik berukuran kecil dan tetap yang mewakili sebuah sinyal audio. Sebuah fungsi *audio fingerprint* harus mampu memetakan sebuah sinyal audio yang memiliki ukuran bit besar menjadi sebuah *audio fingerprint* yang mewakili sinyal audio tersebut dengan ukuran bit kecil.

Audio fingerprint dapat dianalogikan dengan fungsi *hash* dimana sebuah pesan (sinyal audio) akan diproses dengan menggunakan sebuah fungsi hash sehingga dihasilkan sebuah *message digest* atau *digital fingerprint*. Untuk melakukan proses ekstraksi *audio fingerprint* maka sebuah sinyal audio akan pecah menjadi *frame-frame* yang *overlap*.Pada setiap frame akan dilakukan *transform* sehingga diperoleh energinya (*Magnitude*). Hasil transform akan dipecah menjadi 33 kelompok *band*. *Audio*

[1] *Stefanus Irwan Yuanto, Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[2] *Junius Karel T, S.Si., M.T., Dosen Teknik Inforrnatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[3] *Restyandito, S. Kom, MSIS, Dosen Teknik Inforrnatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[4] *P. J. O Doets, R.L. Lagendjik, Theoretical Modeling of a Robust Audio Fingerprinting System, The 2004 IEEE Benelux Signal Processing Symposium.*

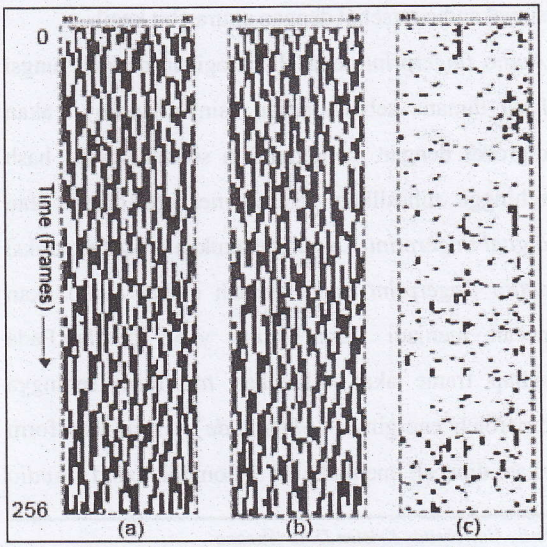
Fingerprint diperoleh dengan menghitung perubahan energi pada *band* dan *frame* yang telah diperoleh dengan rumus sebagai berikut :

$$ED(n,m) = E(n,m) - E(n,m+1) - [E(n-1,m) - E(n-1,m+1)]$$

dengan E(n,m) adalah energi pada *frame* ke n dan kelompok *band* ke m, dan ED(n,m) akan digunakan untuk menentukan bit ke m dari *sub-fingerprint* pada *frame* ke n melalui fungsi F(n,m), yaitu :

$$F(n,m) = \begin{cases} 1 \text{ jika } ED(n,m) > 0 \\ 0 \text{ jika } ED(n,m) \leq 0 \end{cases}$$

Identifikasi audio dilakukan dengan melakukan *tresholding* pada *audio fingerprint* yang akan dibandingkan. Jika F(X) adalah *audio fingerprint* dari obyek audio X dan T adalah nilai *treshold* yang dipilih, maka 2 obyek audio A dan B dikatakan memiliki persamaan jika |F(A) - F(B)| □ T dan dikatakan berbeda jika |F(A) - F(B)| >T. Ilustrasi identifikasi *fingerprint* dapat dilihat pada Gambar 1.



Gambar 1. Ilustrasi Identifikasi *Fingerprint*.
(a) *Fingerprint* Obyek Audio A;
(b) *Fingerprint* Obyek Aaudio B;
(c) Hasil Perbandingan (a) dan (b)

3. Perancangan Sistem

3.1. Struktur Data *Audio Fingerprint*

Hasil ekstraksi *fingerprint* dari sebuah audio akan disimpan dalam bentuk file dengan ekstensi *.afp. Struktur file *.afp dapat dilihat pada Tabel 1.

Offset	Ukuran	Nama Field	Isi
0	4	Chunk ID	Karakter ASCII "WAFP"
4	4	SubChunk1 ID	Karakter ASCII "info"
8	4	SubChunk1 Size	Ukuran isi field AudioInfo
12	N	AudioInfo	Informasi tentang audio (judul, artist, album, dll)
12 + N	4	SubChunk2 ID	Karakter ASCII "data"
16 + N	4	Subchunk2 Size	Ukuran isi field *AudioFingerprint*
20 + N	M	AudioFingerprint	Data *Audio Fingerprint*

Tabel 1. Struktur data file *audio fingerprint* (*.afp)

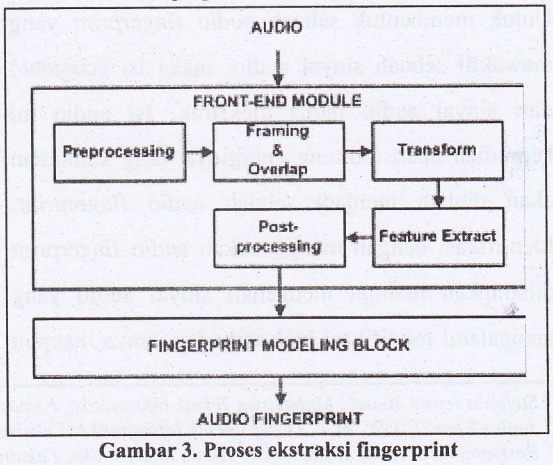
Pada field AudioInfo setiap informasi akan dimulai dengan karakter ASCII "F.M." diikuti dengan nama informasi, karakter ";" dan isi informasinya seperti pada Gambar 2.

"F.M."	Nama Field	";"	Isi Field	"F.M."	Nama Field	";"	Isi Field	... dst

Gambar 2. Struktur penyimpanan field AudioInfo

3.2. Proses Ekstraksi Audio Fingerprint

Proses ekstraksi *audio fingeprint* dapat dibagi menjadi 2 bagian utama seperti yang dapat dilihat pada Gambar 3. yaitu *front-end module* dimana isi(*content*) sebuah sinyal audio akan diekstrak dan bagian *fingerprint modeling-block* dimana isi yang telah diekstrak akan diolah menjadi sebuah *audio fingerprint*.



Gambar 3. Proses ekstraksi fingerprint

3.2.1. Front-end Module

Front-end module terdiri dari 5 bagian sub-proses. Proses pertama adalah *preprocessing*

dimana sinyal akan diubah menjadi mono dengan cara mengambil rata-rata dari keseluruhan *channel* yang ada. Proses kedua adalah *framing&overlap* dimana sinyal akan dipecah menjadi *frame-frame* yang *overlap* dengan panjang *frame* (L) = 370 ms dengan faktor *overlap* 31/32 sehingga ΔL = 11.6 ms dan *frame* yang *overlap* = 358.4 ms. Proses selanjutnya yang dilakukan adalah melakukan *transform* yaitu dengan cara melakukan *Fourier Transform* pada setiap *frame* yang ada dan mengambil *Magnitude*-nya (FFT^2). *Fourier Transform* akan mengubah sebuah sinyal dari kawasan waktu menjadi kawasan frekuensi. Proses keempat adalah *feature extract. Dimana* hasil dari proses *transform* pada setiap frame kemudian akan dibagi menjadi 33 kelompok *band* dimana setiap band akan mewakili *range* frekuensi tertentu. Pembagian band dilakukan berdasar *critical band*[5] dimana frekuensi dibagi menjadi 4 bagian utama yaitu *meat* (20Hz-120Hz), *warmth* (120Hz-2kHz), *treble*(2kHz-10kHz), dan *air*(10kHz-20kHz), dimana bagian *warmth* adalah frekuensi yang paling mudah dibedakan oleh telinga manusia sehingga bagian warmth akan menjadi fokus dari pembagian kelompok *band.* Pembagian kelompok *band* dengan menggunakan *critical band* dapat dilihat pada Tabel 2.

Setelah kelompok *band* diperoleh, maka setiap *band* kemudian akan dilakukan perhitungan energinya, yaitu dengan menjumlahkan *Magnitude* pada setiap kelompok *band* yang ada (E = ∑Magnitude). Proses selanjutnya adalah melakukan *post-processing* dimana satuan energi akan diubah menjadi desibel (dB = 20 * log10(E)).

3.2.2. Fingerprint Modeling-Block

Band	Frekuensi	Band	Frekuensi
1	20 Hz - 40 Hz	18	1.5 kHz - 1.6 kHz
2	40 Hz - 80 Hz	19	1.6 kHz - 1.7 kHz
3	80 Hz - 120 Hz	20	1.7 kHz - 1.8 kHz
4	120 Hz - 200 Hz	21	1.8 kHz - 1.9 kHz
5	200 Hz - 300 Hz	22	1.9 kHz - 2 kHz
6	300 Hz - 400 Hz	23	2 kHz – 2.2 kHz
7	400 Hz - 500 Hz	24	2.2 kHz - 2.6 kHz
8	500 Hz - 600 Hz	25	2.6 kHz - 3 kHz
9	600 Hz - 700 Hz	26	3 kHz - 3.4 kHz
10	700 Hz - 800 Hz	27	3.4 kHz - 4 kHz
11	800 Hz - 900 Hz	28	4 kHz - 5 kHz
12	900 Hz - 1 kHz	29	5 kHz - 6 kHz
13	1 kHz - 1.1 kHz	30	6 kHz - 8 kHz
14	1.1 kHz - 1.2 Hz	31	8 kHz - 10 kHz
15	1.2 kHz - 1.3 kHz	32	10 kHz – 14 kHz
16	1.3 kHz - 1.4 kHz	33	14 kHz - 20 kHz
17	1.4 kHz - 1.5 kHz		

Tabel 2. Pembagian kelompok *band* berdasarkan *critical band*

Pada bagian ini hasil dari *front-end module* akan diubah menjadi *fingerprint* dengan menghitung arah perbedaan energi pada setiap *band* dan *frame* yang telah diperoleh. Perbedaan energi (ED) pada *frame* ke n dan *band* ke m dihitung dengan rumus :

$$ED(n,m) = E(n,m) - E(n,m+1) - [E(n-1,m) - E(n-1,m+1)]$$

sedangkan *sub-fingerprint* (F_n)diperoleh dengan rumus :

$$F(n,m) = \begin{cases} 1 \text{ jika } ED(n,m) > 0 \\ 0 \text{ jika } ED(n,m) \leq 0 \end{cases}$$

dimana nilai 0 menunjukkan bahwa tidak ada perubahan energi atau terjadinya penurunan energi (secara visual akan diwakili oleh titik putih) dan nilai 1 menunjukkan adanya kenaikan energi (secara visual akan diwakili diwakili oleh titik hitam). A*udio fingerprint* adalah gabungan dari seluruh *sub-fingerprint* yaitu F.

3.3. Proses Identifikasi Audio

Proses identifikasi dengan menggunakan *audio fingerprint* dilakukan dengan melakukan *tresholding* terhadap perbandingan *fingerprint*(F)

[5] *iZotope, Mastering with Ozone : Tool, tips and technique , iZotope, Inc, 2004*

dari 2 buah audio dimana nilai treshold yang dipilih adalah 70%. Jadi jika persentase kesamaan *finger-print* dari 2 obyek audio lebih besar atau sama dengan 70%, maka 2 obyek audio tersebut dikatakan sama. Ilustrasi identifikasi dapat dilihat pada Gambar 1.

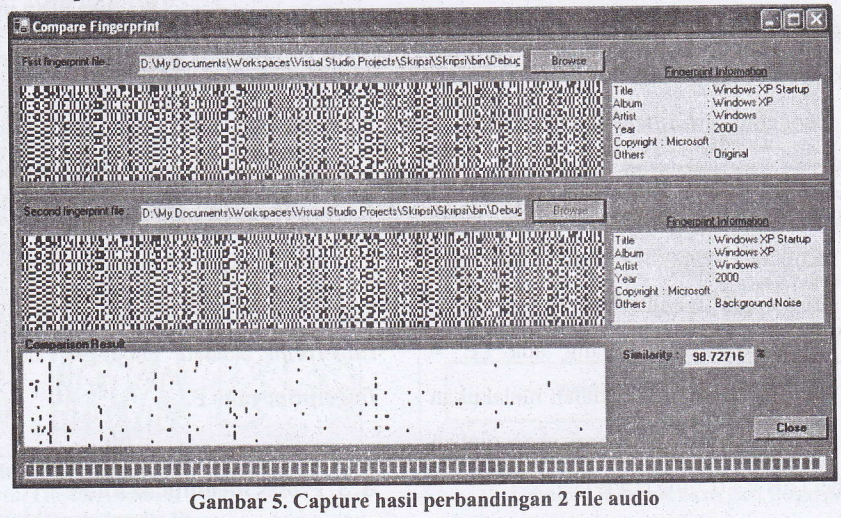
4. Implementasi Sistem

Contoh isi file *audio fingerprint* (*.afp) dapat dilihat pada Gambar 4.

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00000000	57	41	46	50	69	6E	66	6F	00	00	00	79	46	2E	4D	2E	WAFPinfo...yF.M.
00000016	54	69	74	6C	65	3B	57	69	6E	64	6F	77	73	20	58	50	Title;Windows XP
00000032	20	53	74	61	72	74	75	70	46	2E	4D	2E	41	6C	62	75	StartupF.M.Albu
00000048	6D	3B	57	69	6E	64	6F	77	73	20	58	50	46	2E	4D	2E	m;Windows XPF.M.
00000064	41	72	74	69	73	74	3B	57	69	6E	64	6F	77	73	46	2E	Artist;WindowsF.
00000080	4D	2E	59	65	61	72	3B	32	30	30	30	46	2E	4D	2E	43	M.Year;2000F.M.C
00000096	6F	70	79	72	69	67	68	74	3B	4D	69	63	72	6F	73	6F	opyright;Microso
00000112	66	74	46	2E	4D	2E	4F	74	68	65	72	73	3B	4F	72	69	ftF.M.Others;Ori
00000128	67	69	6E	61	6C	64	61	74	61	00	00	05	FC	6D	6D	5D	ginaldata...ümm]
00000144	F0	95	4A	C2	00	6C	A5	6D	D0	82	D2	BB	30	DA	29	45	ð▮JÅ.l¥mÐ▮Ò»0Ú)E
00000160	A0	A4	86	90	00	5B	29	EA	F0	85	6A	91	00	B1	54	6B	¤▮▮.[)êð▮j'.±Tk

Gambar 4. Capture isi file *audio fingerprint* (*.afp)

Contoh perbandingan file audio yaitu startup.wav asli dengan startup.wav yang telah diberi *noise* dapat dilihat pada Gambar 5. Hasil menunjukkan bahwa noise hanya berpengaruh kecil terhadap proses identifikasi dengan hasil persentase kesamaan kedua file adalah 98.72716% (bagian putih) dan persentase perbedaan 1.27284% (bagian hitam).



Gambar 5. Capture hasil perbandingan 2 file audio

Hasil eksperimen yang telah dilakukan untuk perbandingan beberapa file audio (*.wav) asli dengan beberapa variasi perubahan isi file dapat dilihat pada Tabel 3.

Proses	XP Startup.wav	Korg Demo.wav	A-90 Demo.wav
Normalisasi	99.72%	99.79%	99.75%
Mono	55.37%	56.12%	56.43%
Upsampling at 44kHz	86.06%	80.57%	84.56%
Downsampling at 11kHz	83.02%	76.05%	80.18%
Bits Per Sample = 8	55.48%	56.23%	56.51%
Equalization	79.70%	80.89%	76.44%
Distortion	75.80%	67.83%	73.30%
Background Noise	93.37%	87.15%	94.69%
Band Pass Filter (100-1000Hz)	59.15%	58.54%	58.42%
Band Pass Filter (20-10000Hz)	80.23%	78.69%	73.01%
Transpose +1	55.11%	56.46%	56.73%
Reverb	91.39%	88.83%	86.57%

Tabel 3. Hasil uji perbandingan beberapa file audio (*.wav)

Keterangan :

Warna	Persentase Kesamaan
	< 70%
	≥ 70%

5. Kesimpulan

Dengan memperhatikan keseluruhan proses dan hasil eksperimen yang telah dilakukan, maka penulis dapat menarik kesimpulan sebagai berikut:

1. Dengan ukurannya yang kecil, sebuah *audio fingerprint* mampu mewakili sebuah sinyal audio sehingga dapat mempercepat proses pencarian atau identifikasi
2. Teknologi *audio fingerprint* mampu mengenali sinyal audio yang mengalami perubahan

kualitas karena *noise* dan perubahan format penyimpanan

3. Perubahan pada sinyal yang membuat perubahan frekuensi secara besar akan menurunkan tingkat kesamaan sinyal tersebut dengan sinyal aslinya

4. Sistem tidak mampu mengenali perbedaan pitch, tempo, dan terjadinya shifting pada sinyal audio.

Daftar Pustaka

Cano, Pedro, dkk, (2005), *Audio Fingerprinting : Concept and Application*, http://www.springerlink.com, diakses tanggal 10 Maret 2006.

Cormen, Leiserson, Rivest & Stein, *Introduction to Algoritmhs Second Edition*, Mc Graw-Hill Companies.

Doets , P.J.O, R.L. Lagendijk (2004), *Theoretical Modeling of a Robust Audio Fingerprinting System*, The 2004 IEEE Benelux Signal Processing Symposium.

Gruhne, Matthias, dkk, (2003), *Robust Audio Identification for Commercial Applications*, http://VirtualGoods.tu-ilmenau.de/2003/ Robust_Audio_Identification_for_Commercial_Applications.pdf, diakses tanggal 10 Maret 2006.

iZotope, *Mastering with Ozone : Tool, tips and technique* , iZotope, Inc, 2004.

Relisoft, *Fourier Transform*, http://www.relisoft.com, diakses tanggal 23 Februari 2007.

Roman Kuc, *Introduction to Digital Signal Processing*, Singapore, Mc-Graw-Hill Book Co, 1982.

Smith, Julius O, (2003) *Mathematics of the Discrete Fourier Transform (DFT), with Music and Audio Applications*, http://ccrma-www.stanford.edu/~jos/r320/, diakses tanggal 27 April 2006.

Sonic Spot, *WAVE File Format*, http://www.sonicspot.com/guide/wavefiles.html, diakses tanggal 17 Februari 2006.

Wilson, Scott, (2003) *WAVE PCM soundfile format*, http://ccrma.stanford.edu/courses/422 /projects/WaveFormat/, diakses tanggal 17 Februari 2006.

Stefanus Irwan Yuanto
Crown_acts@yahoo.com
Jl. M.T. Haryono 51
Temanggung 56213 Jawa Tengah