

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

DIGITAL SIGNAL PROCESSING

Laporan ini merupakan bagian dari Assignment EL5226 – Jaringan Informasi Sosial ITB2015. Source code dan laporan juga tersedia di link github : [https://github.com/soedomoto/ITB2015/tree/EL5226/Assignment 2 - DSP](https://github.com/soedomoto/ITB2015/tree/EL5226/Assignment%202%20-%20DSP)

1. AUDIO FINGERPRINTING AND RECOGNITION

Merekam suara dan mencocokkan dengan potongan suara yang tersedia untuk identifikasi

Audio fingerprinting merupakan suatu teknik identifikasi suara dengan cara memecahnya menjadi beberapa sampel. Sebuah file audio dikonversi menjadi grafik sinus, kemudian sampel diperoleh dengan cara pencarian titik puncaknya. Algoritma pencarian titik puncak ini dikenal dengan FFT (https://en.wikipedia.org/wiki/Fast_Fourier_transform). Misalnya sebuah file .mp3 stereo (2 channel) dengan durasi 3 menit akan memiliki sampel :

$$3 \text{ min} * 60 \text{ sec} * 44100 \text{ samples per sec} * 2 \text{ channels} = 15,876,000 \text{ samples}$$

Kemudian sebanyak 15,876,000 samples tersebut akan disimpan ke dalam sebuah database. Begitu seterusnya untuk setiap file audio yang akan diidentifikasi. Kemudian ketika terdapat sebuah potongan audio, misalnya 10 detik audio .mp3 ingin diidentifikasi siapakah penyanyi dari potongan audio tersebut, maka potongan audio tersebut akan di-sampling, kemudian setiap sampling akan dicocokkan pada database fingerprint. Metode seperti ini seperti yang diterapkan pada beberapa aplikasi pencarian lagu populer seperti Midomi (<http://www.midomi.com/>).

Salah satu tools atau implementasi audio fingerprinting and recognition adalah Dejavu (<https://github.com/worldveil/dejavu>). Dejavu merupakan aplikasi berbasis python yang bekerja dengan algoritma FFT. Dejavu bekerja dengan memanfaatkan database MySQL, akan tetapi untuk meningkatkan portabilitas, sebuah kelas python yang mengimplementasikan interface dejavu.database.Database dibuat agar mendukung database SQLite3 ([dejavu_sqlite.py](#)). Berikut penjelasan singkatnya :

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

System Dependencies :

Aplikasi dejavu merupakan aplikasi yang ditulis dengan bahasa Python, sehingga sebenarnya dejavu bersifat platform independent. Tapi dalam kasus ini, dejavu akan dijalankan pada platform berbasis Linux Ubuntu 14.04 Trusty Tahr. Dejavu memerlukan beberapa dependency terhadap aplikasi tertentu :

1. ffmpeg, untuk konversi audio ke .wav format,
2. portaudio, untuk membaca file audio, termasuk membaca audio lewat microphone,
3. audacity, untuk modifikasi file audio (join, splice, resample)

Python Dependencies :

1. numpy, untuk membaca signal audio,
2. pydub, python wrapper ffmpeg,
3. matplotlib, untuk menggambar spectrogram dan plotting,
4. scipy, untuk pencarian titik puncak dengan algoritma FFT,
5. pyaudio, python wrapper untuk portaudio

Penjelasan Program :

Download [Dejavu](#), atau modifikasinya dengan menghilangkan dependency terhadap MySQL disini [Modifikasi Dejavu](#). Dejavu yang telah dimodifikasi memiliki dependency terhadap [dejavu_sqlite.py](#). Struktur database yang digunakan dalam dejavu-sqlite adalah :

```
CREATE TABLE songs (  
    song_id integer primary key autoincrement,  
    song_name varchar(255) not null,  
    fingerprinted tinyint default 0,  
    file_shal binary(10) not null  
);
```

```
CREATE TABLE fingerprints (  
    hash binary(10) not null,  
    song_id unsigned mediumint not null,  
    offset unsigned int not null,  
    UNIQUE(song_id, offset, hash)  
);
```

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsional Layanan Teknologi Informasi

Import library yang diperlukan :

```
from dejavu import Dejavu
from dejavu.recognize import FileRecognizer, MicrophoneRecognizer
import dejavu_sqlite
```

Buat konfigurasi database dengan menggunakan sqlite :

```
config = {
    "database_type": "sqlite",
    "database": {
        "db": "dejavu-test.db"
    }
}
```

Instansiasi kelas Dejavu :

```
djv = Dejavu(config)
```

Fingerprint semua file .mp3 dalam sebuah directory (File audio tersedia [disini](#)) :

```
djv.fingerprint_directory("mp3", [".mp3"])
```



```
dejavu/fingerprint.py:82: RuntimeWarning: divide by zero encountered in log10
  arr2D = 10 * np.log10(arr2D)
dejavu/fingerprint.py:82: RuntimeWarning: divide by zero encountered in log10
  arr2D = 10 * np.log10(arr2D)
Finished channel 1/2 for mp3/Brad-Sucks--Total-Breakdown.mp3
Fingerprinting channel 2/2 for mp3/Brad-Sucks--Total-Breakdown.mp3
Finished channel 1/2 for mp3/Sean-Fournier--Falling-For-You.mp3
Fingerprinting channel 2/2 for mp3/Sean-Fournier--Falling-For-You.mp3
Finished channel 1/2 for mp3/Josh-Woodward--I-Want-To-Destroy-Something-Beautiful.mp3
Fingerprinting channel 2/2 for mp3/Josh-Woodward--I-Want-To-Destroy-Something-Beautiful.mp3
Finished channel 2/2 for mp3/Brad-Sucks--Total-Breakdown.mp3
Finished channel 1/2 for mp3/Choc--Eigenvalue-Subspace-Decomposition.mp3
Fingerprinting channel 2/2 for mp3/Choc--Eigenvalue-Subspace-Decomposition.mp3
Fingerprinting channel 1/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3
Finished channel 2/2 for mp3/Sean-Fournier--Falling-For-You.mp3
Finished channel 2/2 for mp3/Josh-Woodward--I-Want-To-Destroy-Something-Beautiful.mp3
Finished channel 1/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3
Fingerprinting channel 2/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3
Finished channel 2/2 for mp3/Choc--Eigenvalue-Subspace-Decomposition.mp3
Finished channel 2/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3
>>>
```

Deteksi potongan lagu (minimal 5 detik potongan lagu), dan kenali penyanyi-nya (File potongan lagu untuk testing tersedia [disini](#)) :

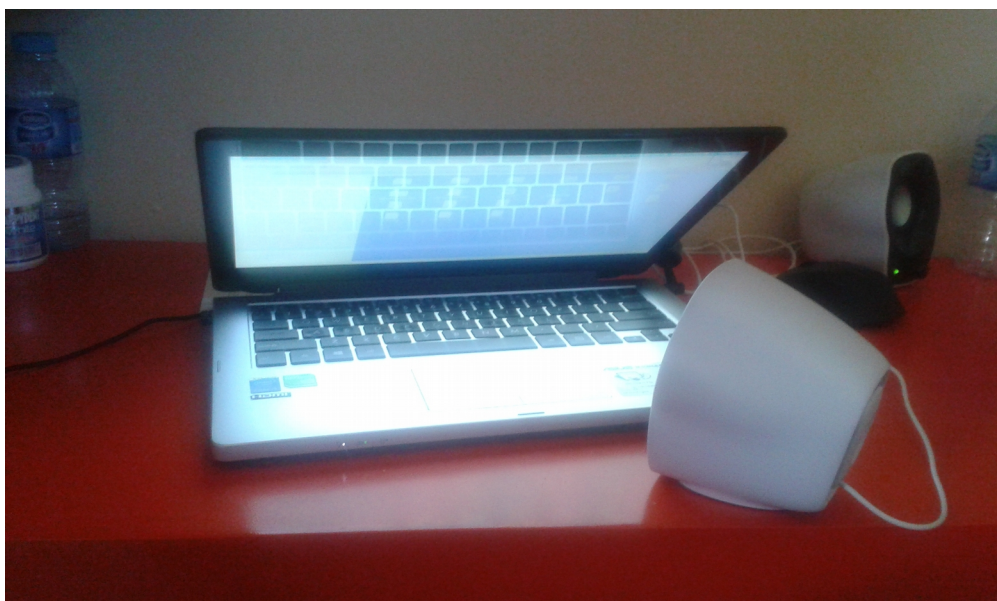
```
song = djv.recognize(FileRecognizer, "test/test1.mp3")
print "From file we recognized: %s\n" % song
```

Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

```
Terminal
>>> config = {
...     "database_type": "sqlite",
...     "database": {
...         "db": "/home/soedomoto/Documents/dejavu-test.db"
...     }
... }
>>> djv = Dejavu(config)
database opened successfully
>>> song = djv.recognize(FileRecognizer, "test/test1.mp3")
>>> print("From file we recognized: %s\n" % song)
From file we recognized: {'song_id': 2, 'song_name': 'Sean-Fournier--Falling-For-You', 'file_sha1': 'a9d18b9b9daa467350d1b6b249c36759282b962e', 'confidence': 917, 'offset_seconds': 63.20472, 'match_time': 1.6062519550323486, 'offset': 1361}
>>>
>>> rtwarnings
>>> rtjson
>>> ngs.filterwarnings("ignore")
>>>
>>> dejavu import Dejavu
>>> dejavu.recognize import FileRecognizer, MicrophoneRecognizer
>>>
>>> █ config from a JSON file (or anything outputting a python dictionary)
```

Deteksi potongan lagu melalui microphone, dan kenali penyanyi-nya (File potongan lagu untuk testing tersedia [disini](#)) :

```
secs = 5
song = djv.recognize(MicrophoneRecognizer, seconds=secs)
if song is None:
    print "Nothing recognized -- did you play the song out loud so your mic could hear it? :)"
else:
    print "From mic with %d seconds we recognized: %s\n" % (secs, song)
```



Nama : Aris Prawisudatama
NIP : 23215131
Jurusan : Teknik Elektro
Opsi Layanan Teknologi Informasi

```
Terminal
>>>
>>>
>>> secs = 5
>>> song = djv.recognize(MicrophoneRecognizer, seconds=secs)
ALSA lib pcm_dsnoop.c:618:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1022:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
bt_audio_service_open: connect() failed: Connection refused (111)
bt_audio_service_open: connect() failed: Connection refused (111)
bt_audio_service_open: connect() failed: Connection refused (111)
bt_audio_service_open: connect() failed: Connection refused (111)
ALSA lib pcm_dmix.c:1022:(snd_pcm_dmix_open) unable to open slave
>>> if song is None:
...     print "Nothing recognized -- did you play the song out loud so your mic
could hear it? :)"
... else:
...     ("From mic with %d seconds we recognized: %s\n" % (secs, song))
...
"From mic with 5 seconds we recognized: {'song_id': 5, 'song_name': 'The-Lights-
Galaxia--While-She-Sleeps', 'file_sha1': '355223ebf3ee43eed0f45eca688faad186fc31
67', 'confidence': 3, 'offset_seconds': 32.18286, 'offset': 693}\n"
>>>
```

2. FACE DETECTION

Merekam video muka dan mengamati pergerakannya (kanan, kiri)