

## EKSTRAKSI PITCH INTERVAL DARI SINYAL SENANDUNG UNTUK IDENTIFIKASI LAGU

Wahyu Kusuma dan Irwan Arifin

Jurusan Sistem Informasi, Fakultas Ilmu Komputer Universitas Gunadarma  
Jl. Margonda Raya 100, Pondok Cina, Depok, 16424  
E-mail: {wahyukr, irwan}@staff.gunadarma.ac.id

### ABSTRAKSI

Pencarian lagu dari suatu basis data dapat dilakukan melalui kueri teks judul lagu dan nama penyanyi. Pencari lagu terkadang tidak mengetahui atau lupa akan judul lagu atau nama penyanyinya, melainkan hanya mengetahui potongan lagu saja dalam bentuk senandung. Dalam keadaan seperti ini, diperlukan suatu sistem yang dapat mengekstraksi informasi dari potongan lagu tersebut, untuk kemudian digunakan untuk pencarian lagu. Pada penelitian ini dikembangkan sebuah aplikasi yang berfungsi untuk mengekstraksi ciri pitch-interval, dan bukannya not lagu, karena ciri ini invarian terhadap variasi tangga nada. Data masukan aplikasi ini menjalani beberapa proses, sebelum akhirnya menghasilkan fitur pitch interval. Proses ini antara lain deteksi onset senandung (terdiri dari proses *rectifying*, *enveloping* menggunakan filter, *diferensiasi*, dan *thresholding*). Posisi onset yang diperoleh digunakan untuk mensegmentasi sinyal, sehingga setiap segmen hanya berisi satu nada. Selanjutnya setiap segmen menjalani proses *pitch-tracking* melalui analisis frekuensi dengan bantuan transformasi *fourier* dan *fuzzy logic*, sehingga akhirnya diperoleh keluaran berupa sekuens *pitch*. Sekuens *pitch* selanjutnya diubah menjadi *pitch interval* melalui proses *diferensiasi*. Keluaran sistem dari sejumlah masukan senandung diujicobakan dalam proses identifikasi yang menggunakan ANN. Ujicoba yang dilakukan menunjukkan tingkat keberhasilan di atas 80%.

**Kata kunci:** *pitch interval*, *senandung (humming)*, *analisis frekuensi*, ANN.

### 1. PENDAHULUAN

Pencarian lagu dari suatu basis data dapat dilakukan melalui kueri teks judul lagu dan nama penyanyi. Pencari lagu terkadang tidak mengetahui atau lupa akan judul lagu atau nama penyanyinya, melainkan hanya bisa menyenandungkan potongan lagu yang dimaksud. Dalam interaksi antar manusia, senandung dari pencari lagu sudah bisa dijadikan bahan masukan bagi lawan bicara untuk memulai proses pencarian lagu dalam ingatannya. Kemampuan manusia dalam memulai proses pencarian lagu berdasarkan masukan senandung ini menarik untuk diimplementasikan ke dalam komputer. Menjadi menarik karena jika kemampuan ini sudah dimiliki komputer, kita bisa memilih lagu mp3 di komputer hanya dengan melakukan senandung tanpa harus masuk ke prosedur pemilihan file. Kemampuan ini juga membuka kemungkinan pencarian lagu di internet melalui senandung.

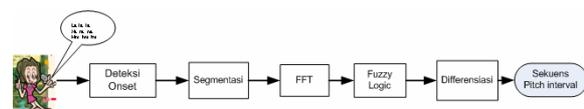
Masalah yang diangkat dalam penelitian ini adalah bagaimana cara memproses senandung sehingga bisa digunakan dalam pencarian data lagu. Penelitian ini hanya dibatasi pada ekstraksi karakteristik senandung hingga diperoleh fitur spesifik yang bisa menjadi ciri suatu lagu. Hasil dari penelitian ini nantinya dapat diintegrasikan ke dalam basis data lagu dengan indeks yang sesuai.

Proses ekstraksi pada penelitian ini berlangsung secara *off-line*. Data sampel berbentuk

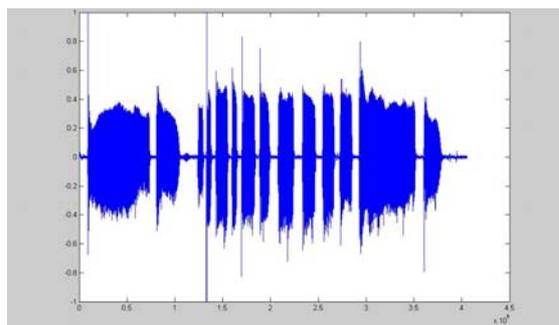
file suara berformat WAV (sampling rate 44,1 kHz, mono). Sebagian data sampel dibuat menggunakan perangkat MIDI, dengan maksud agar terdapat acuan yang bisa digunakan untuk pengukuran kesalahan sistem. Data sampel lain dibuat dengan merekam senandung dalam berbagai variasi lagu, penenandung (pria/wanita), dan bentuk senandung (na na, la la, dan siulan).

### 2. METODOLOGI, HASIL DAN PEMBAHASAN

Gambar 1 berikut ini menunjukkan rangkaian proses yang digunakan untuk mengekstraksi interval *pitch* dari sinyal musik. Seluruh proses dibuat menggunakan perangkat lunak MATLAB. Gambar 2 memperlihatkan satu contoh sinyal hasil senandung, yang akan menjalani proses. Proses diawali dengan deteksi onset, untuk melacak posisi-posisi waktu dimana suatu not mulai berbunyi.



Gambar 1. Blok diagram tahapan ekstraksi pitch interval



**Gambar 2.** Contoh sinyal masukan hasil senandung (refrain lagu *Wind of Change* - Scorpion)

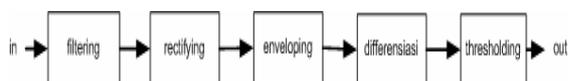
Proses deteksi onset sendiri terdiri atas beberapa subproses seperti tergambar pada gambar 3 di bawah ini. Inti dari proses deteksi onset ini adalah mencari posisi-posisi gradien yang bernilai tinggi (di atas ambang yang ditentukan) dari kontur sinyal. Pada penelitian yang dilakukan, blok filtering masih bersifat opsional, mengingat sinyal berasal dari musik monofonik. Proses *rectifying* adalah proses penyearahan gelombang, dimana seluruh simpangan yang bernilai negatif dijadikan positif. Untuk itu cukup digunakan operator absolut pada proses ini.

$$\text{Rectifying: } y(n) = |x(n)| \quad (1)$$

Sinyal yang telah menjalani proses *rectifying*, selanjutnya dimasukkan pada proses *enveloping* untuk membentuk kontur sinyal. Di sini proses *enveloping* dilakukan melalui proses konvolusi sinyal dengan *mask* filter.

$$\text{Konvolusi: } x(n) * z(n) = \sum_{m=0}^{M-1} x(m)z(n-m) \quad (2)$$

Konvolusi dilakukan dengan nilai M sebesar 1000.



**Gambar 3.** Blok Diagram deteksi onset

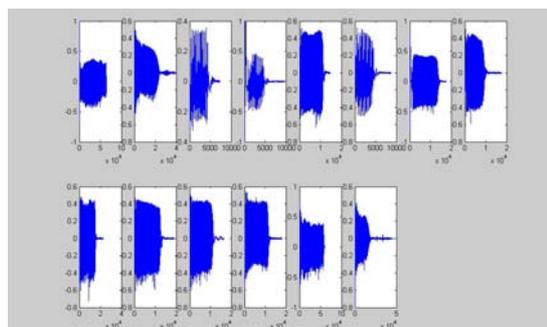
Amplop sinyal yang menggambarkan kontur sinyal, selanjutnya menjalani proses diferensiasi untuk mendapatkan nilai gradien.

$$\text{Diferensiasi: } y(n) = x(n+1) - x(n) \quad (3)$$

dengan  $n = 1 \dots (N-1)$  dan  $N$  adalah jumlah titik sampel. Dari diferensiasi ini diperoleh keluaran berupa nilai-nilai gradien. Untuk mendapatkan nilai gradien yang tinggi, seluruh nilai dibandingkan dengan nilai ambang pada proses *thresholding*. Gradien dengan nilai di bawah ambang dijadikan nol, dan gradien dengan nilai di atas ambang tetap

dipertahankan. Posisi-posisi dari gradien yang nilainya di atas ambang merupakan posisi onset dari not.

Posisi onset yang diperoleh selanjutnya digunakan pada proses berikutnya, yaitu segmentasi. Pada proses segmentasi ini, sinyal dipotong-potong sehingga pada setiap potongan (segmen) sinyal akan terdapat satu not (gambar 4).



**Gambar 4.** Sekuens not hasil keluaran proses segmentasi

Sekuens segmen kemudian dikirim ke proses *pitch tracking* menggunakan FFT. Dari spektrum FFT yang diperoleh, dicari nilai frekuensi yang memiliki *magnitude* terbesar. Nilai frekuensi ini kemudian dikonversi menjadi kode not dengan bantuan *fuzzy logic*. Penggunaan *fuzzy logic* dimaksudkan agar dalam proses konversi frekuensi menjadi not, sistem memiliki toleransi yang memadai terhadap pergeseran frekuensi pada not (fals).

Dalam prosesnya, *fuzzy logic* akan menghasilkan keluaran berupa kode not dengan rentangan dua oktaf. Interpretasi kode not menjadi not yang sebenarnya mengacu pada tabel 1 berikut ini.

**Tabel 1.** Kode not

<b>Not</b>	C4	C4#	D4	D4#	E4	F4	F4#
<b>Kode</b>	1	1.5	2	2.5	3	3.5	4
<b>Not</b>	G4	G4#	A4	A4#	B4		
<b>Kode</b>	4.5	5	5.5	6	6.5		

<b>Not</b>	C5	C5#	D5	D5#	E5	F5	F5#
<b>Kode</b>	7	7.5	8	8.5	9	9.5	10
<b>Not</b>	G5	G5#	A5	A5#	B5		
<b>Kode</b>	10.5	11	11.5	12	12.5		

Di akhir proses, pada keluaran *fuzzy logic* akan terdapat sekuens not. Sekuens ini masih sangat bergantung pada tangga nada yang digunakan pada saat pengguna bersenandung. Fitur ini harus disederhanakan untuk menghemat penggunaan ruang di basis data. Untuk itu, ketergantungan sekuens not terhadap tangga nada dihilangkan melalui proses diferensiasi. Dari proses

diferensiasi inilah akan diperoleh sekuens interval *pitch* yang akan dijadikan petunjuk dalam pencarian lagu di basis data.

Terdapat beberapa cara untuk menggunakan interval *pitch* dalam proses pencarian lagu. Selain bisa dijadikan data masukan untuk kueri, interval *pitch* juga bisa dijadikan masukan untuk Jaringan Syaraf Tiruan. Pada penelitian ini interval *pitch* dicoba digunakan untuk mencari judul lagu dengan bantuan JST. JST didisain berarsitektur *feedforward* dengan 15 neuron pada *input layer*, yang berarti dapat menampung maksimal 15 data interval *pitch* dari satu sekuens. *Hidden layer* terdiri atas 25 neuron dan *output layer* terdiri atas 10 neuron sebagai representasi vektor 10 bit. *Hidden* dan *output layer* menggunakan fungsi aktivasi sigmoid. Metode belajar yang digunakan adalah Levenberg Marquardt (LM). Sebagai data pelatihan, digunakan sampel data interval *pitch* yang berasal dari 10 lagu: *Bengawan Solo*, *Maju Tak Gentar*, *Selendang Sutra*, *Wind of Change*, *Gambang Suling*, *Halo-Halo Bandung*, *Gundul Pacul*, *Kolam Susu*, *Satu Nusa Satu Bangsa*, *Hey Jude*. Data interval *pitch* dari kesepuluh lagu ini diperoleh dengan terlebih dahulu memainkan potongan lagu pada perangkat MIDI, kemudian sistem menghitung interval *pitch* yang terjadi. Pelatihan berjalan dengan baik, dimana JST dapat mencapai target MSE:  $1 \times 10^{-3}$ .

Sebagai data uji, dibuat senandung dari sepuluh lagu di atas, dengan variasi suara wanita-pria, bentuk senandung (na-na, la la, siul). Data uji juga dilengkapi dengan senandung artifisial yang dibuat menggunakan MIDI, dengan suara *Voice Ooh*, *Choir Aahs*, *Synthetic Voice*.

Hasil pengujian menunjukkan bahwa sistem dapat mengidentifikasi lagu dari senandung dengan tingkat kebenaran 80%. Tingkat kebenaran sangat tinggi pada saat digunakan suara yang berasal dari perangkat MIDI. Sementara itu, kendala banyak ditemui pada saat pengidentifikasian lagu dengan input senandung suara manusia. Masalah yang ditemukan pada senandung manusia antara lain terjadinya inkonsistensi *pitch*, dimana deviasi frekuensi nada fals bernilai cukup besar, sehingga sistem salah dalam mengidentifikasi not yang dimaksud. Kesalahan dalam identifikasi not ini juga menimbulkan kesalahan pada penghitungan interval *pitch*. Kesalahan yang lain timbul dari gaya senandung, yang tanpa disadari menimbulkan efek *bending*. Not yang disenandungkan dengan efek *bending* oleh sistem tetap dinyatakan sebagai satu not, namun ini menjadi masalah pada saat proses *pitch tracking* dijalankan. Akibatnya, frekuensi yang keluar bukanlah frekuensi not yang dimaksud.

Masalah dalam pengujian juga muncul dari keterbatasan rentangan not yang dimiliki oleh *fuzzy logic* yang hanya dua oktaf. Masalah muncul jika

tanpa disadari senandung sudah keluar dari batasan frekuensi yang tercakup. Akibatnya not tidak teridentifikasi sehingga proses identifikasi lagu secara keseluruhanpun menjadi salah.

Variasi senandung berupa na-na, la-la tidak memberikan efek negatif selama pelantunannya dilakukan secara datar dan terkontrol (diusahakan tidak fals).

### 3. KESIMPULAN DAN SARAN

Secara keseluruhan telah berhasil dibuat metode untuk mengekstraksi interval *pitch* dari data masukan berupa sinyal analog berisi senandung. Dalam pengaplikasiannya, metode ini memerlukan perhatian dalam beberapa hal, yaitu:

- ✓ senandung harus benar-benar berada dalam interval not yang dicakup oleh sistem
- ✓ senandung tidak boleh mengandung efek *bending*
- ✓ senandung harus diusahakan dilakukan dengan *pitch control* yang baik

Beberapa hal bisa dilakukan untuk mengembangkan kemampuan sistem yang telah dibuat dalam penelitian ini. Di antaranya adalah memperluas cakupan not *fuzzy logic* sehingga dapat melingkupi seluruh not dan *pitch* yang mungkin ditimbulkan oleh senandung dan siulan. Keluaran dari sistem dapat diintegrasikan pada basis data untuk tujuan kueri. Kueri yang dilakukan sebaiknya bersifat fleksibel, yang cenderung mengandalkan 'kemiripan' dibandingkan 'kesamaan'.

### DAFTAR PUSTAKA

- Ghias, A., Logan, J., Chamberlin, D., and Smith, B., 1995, *Query by humming: Musical information retrieval in an audio database*. In *ACM Multimedia 1995*, pages 231-236
- Gutiérrez, and Gómez, E., 2002, *Melodic Description of Audio Signals For Music Content Processing*, Research Work, PhD Program Informática i Comunicació digital, Universitat Pompeu Fabra, Barcelona.
- Hagan, Martin T., 1996, *Neural Network Design*, PWS Publishing Company, Boston
- Hashiguchi, H., Nishimura, T., Takita, J., Zhang, J.X., and Oka, R., 2001, *Music Signal Spotting Retrieval by Humming Query Using Model Driven Path Continuous Dynamic Programming*, *SCI*.
- Klapuri, A., 2001, *Automatic transcription of musical recordings*, In *Consistent & Reliable Acoustic Cues Workshop*
- Lu, L., Hong You, Zhang, H., 2001, *A NEW APPROACH TO QUERY BY HUMMING IN MUSIC RETRIEVAL*, in *ICME 2001 Conference proceedings*.

- Nishimura, T., Hashiguchi, H., Takita, J., 2001, *Music Signal Spotting Retrieval by a Humming Query Using Start Frame Feature Dependent Continuous Dynamic Programming*, In *Proceedings of International Symposium on Music Information Retrieval*, Bloomington, IN, USA.
- Proakis, JG, et al, 1992, *Advanced Digital Signal Processing*, Macmillan Publishing Company, New York
- Zhu, Y., Shasha, D., 2003, *Warping Indexes with Envelope Transforms for Query by Humming*, in SIGMOD 2003, June 9-12, San Diego, CA.