QUERY BY HUMMING MUSIC INFORMATION RETRIEVAL DENGAN MENGGUNAKAN EKSTRAKSI MELODI BERBASIS DNN-LSTM DAN FILTRASI DERAU

TESIS

Karya tulis sebagai salah satu syarat
untuk memperoleh gelar Magister dari
Institut Teknologi Bandung

Oleh
ANDREAS NOVIAN DWI TRIASTANTO
NIM: 23518002
(Program Studi Magister Informatika)

INSTITUT TEKNOLOGI BANDUNG
November 2020

QUERY BY HUMMING MUSIC INFORMATION RETRIEVAL DENGAN MENGGUNAKAN EKSTRAKSI MELODI BERBASIS DNN-LSTM DAN FILTRASI DERAU

Oleh
ANDREAS NOVIAN DWI TRIASTANTO
NIM: 23518002
(Program Studi Magister Informatika)

Institut Teknologi Bandung

Menyetujui

Dosen Pembimbing

Tanggal 5 November 2020

Ir. Rila Mandala M.Eng.,Ph.D.
(NIP: 19680803 199302 1 001)

DAFTAR ISI

DAFTAR GAMBAR DAN ILUSTRASI

DAFTAR TABEL

Bab I Pendahuluan

I.1 Latar Belakang

Teknologi mesin pencari sudah menjadi kebutuhan sehari-hari yang sulit untuk dilepaskan. Budaya mencari informasi dengan bertanya ke orang lain perlahan sudah mulai berganti dengan mencari sendiri di mesin pencari melalui ponsel pintar masing-masing. Saat ini, pencarian informasi paling umum dilakukan dengan menggunakan *query* berupa teks atau gambar. Untuk sebagian besar kebutuhan, dua jenis *query* tersebut sudah cukup sebagai bahan untuk mesin pencari memberikan informasi yang dibutuhkan. Namun, untuk beberapa kasus khususnya pencarian musik, dua jenis *query* tersebut kurang cocok digunakan.

Ghias dkk. (1995) menyatakan bahwa cara paling efektif dan natural untuk melakukan pencarian musik adalah dengan menggumamkan lagu yang ingin dicari atau disebut juga dengan *query by humming*. Ghias dkk. (1995) pertama kali memperkenalkan teknik ini menggunakan metode *pitch tracking* untuk merepresentasikan konten melodi dari sebuah lagu dan menggunakan *pattern matching* untuk melakukan pencarian ke basis data. Sejak itu, pengembangan dari teknik ini banyak dilakukan untuk meningkatkan akurasi dan kecepatan pencarian musik dengan *query* berupa gumaman.

Beberapa penelitian terbaru untuk topik ini sudah berhasil mencapai nilai akurasi di atas 80% dengan waktu pencarian di bawah 5000 ms, seperti yang dilakukan oleh Zhou dkk. (2017). Penelitian ini membandingkan beberapa teknik *matching* atau pencocokan seperti *Dynamic Time Warping* (DTW)*, Hidden Markov Model* (HMM)*, Location Sensitive Hash* (LSH) dan kombinasi antara DTW dan LSH. Makarand dan Parag (2018) memperkenalkan *unified algorithm* yang menggabungkan algoritma *mode normalized frequency* (MNF) menggunakan *edit distance* dan n-gram. Hasil dari penelitian ini diukur menggunakan *mean reciprocal rank* (MRR) dan mendapatkan hasil terbaik 0.59.

Kedua penelitian tersebut sama-sama menggunakan teknik untuk melakukan pencarian kepada basis data yang sudah dikecilkan jumlahnya terlebih dahulu. Perbedaannya adalah jika pada penelitian Zhou dkk. (2017) proses pengecilan basis data dilakukan dengan melakukan pencocokan menggunakan algoritma yang menghasilkan akurasi *top-twenty* yang bagus sementara pada penelitian Makarand dan Parag (2018) digunakan indeks terbalik (*inverted index*) yang sudah dikomputasi sebelumnya (*precomputed*) pada basis data. Makarand dan Parag (2018) tidak menyebutkan secara eksplisit waktu yang dibutuhkan untuk melakukan pencarian dengan algoritmanya namun proses pengecilan basis data dengan indeks terbalik yang sudah dikomputasi sebelumnya dirasa lebih cepat dibandingkan melakukan dua algoritma pencocokan setiap kali akan melakukan pencarian.

Pencarian musik dengan *query* berupa gumaman merupakan bagian dari domain penelitian sistem temu balik informasi untuk musik (*music information retrieval*). Beberapa penelitian terbaru pada domain ini berfokus pada teknik ekstraksi melodi untuk mendapatkan representasi melodi yang terbaik dari sebuah rekaman musik. Teknik ini juga merupakan salah satu teknik yang dibutuhkan untuk membangun sistem *query by humming*. Salah satu penelitian terbaru pada topik ini dilakukan oleh Cao dkk. (2020) yang menggunakan pendekatan pembelajaran dengan tugas ganda (*multi-task learning approach*) berbasis *deep neural network* (DNN) dan *long short term memory* (LSTM). Teknik ini melakukan pendekatan pembelajaran terhadap dua tugas sekaligus, yaitu estimasi nada (*pitch estimation*) dan deteksi suara (*voicing detection*). Pendekatan ini terbukti dapat menghasilkan akurasi yang lebih tinggi dan kemampuan generalisasi yang lebih baik.

Menurut Makarand dan Parag (2018), salah satu teknik yang bisa ditambahkan pada sistem *query by humming* dengan harapan dapat meningkatkan akurasinya adalah teknik untuk mereduksi derau (*noise*). Teknik ini digunakan sebelum proses ekstraksi melodi. Beberapa peneliti belum menerapkan teknik ini dalam pembuatan sistem *query by humming* dan sengaja memilih untuk menggunakan audio yang berderau sebagai *query* dan basis data supaya merepresentasikan kondisi di dunia

nyata. Teknik untuk mereduksi derau yang sudah dibuat pada umumnya digunakan untuk mereduksi derau pada sinyal yang berisi ucapan (*speech*). Rajini dkk. (2019) melakukan perbandingan pada beberapa teknik untuk melakukan reduksi derau yang dapat dikelompokkan menjadi dua metode, yaitu metode filtrasi dan *neural network*. Penelitian tersebut menghasilkan kesimpulan berupa metode filtrasi lebih baik dalam melakukan reduksi derau pada sinyal ucapan dibandingkan dengan metode *neural network* yang lebih kompleks dan membutuhkan waktu eksekusi yang lebih lama. Siam dkk. (2019) memperkenalkan salah satu teknik filtrasi untuk melakukan peningkatan kualitas sinyal suara dengan *Fourier series decomposition* dan *spectral subtraction* yang sudah dicoba untuk domain identifikasi pembicara. Penelitian ini menghasilkan kesimpulan bahwa teknik yang diajukan memiliki hasil yang lebih baik dibandingkan beberapa teknik lain, yaitu *Wiener filter* dan *spectral subtraction*, dan terbukti dapat digunakan untuk menghasilkan sistem yang *robust* pada domain identifikasi pembicara.

I.2 Masalah Penelitian

Seperti yang telah disampaikan sebelumnya, teknik filtrasi untuk mereduksi derau pada sistem *query by humming* dipercaya dapat meningkatkan akurasi pencarian. Selain itu, penggunaan teknik ekstraksi melodi yang lebih baik juga dirasa dapat menambah akurasi dan membuat sistem mampu menangani jenis musik yang lebih beragam. Oleh karena itu, masalah yang diangkat untuk penelitian ini adalah:

1. Bagaimana performansi sistem *query by humming* dengan menggunakan teknik ekstraksi melodi berbasis DNN-LSTM?
2. Bagaimana performansi sistem *query by humming* dengan menggunakan teknik reduksi derau *Fourier Series Decomposition* dan *Spectral Subtraction*?
3. Bagaimana performansi sistem *query by humming* dengan menggunakan teknik ekstraksi melodi berbasis DNN-LSTM dan teknik reduksi derau *Fourier Series Decomposition* dan *Spectral Subtraction*?

I.3 Tujuan Penelitian

Tujuan yang ingin didapatkan dari penelitian ini adalah:

1. Membandingkan performansi sistem *query by humming* buatan Makarand dan Parag (2018) dengan penggunaan teknik ekstraksi melodi berbasis DNN-LSTM pada sistem yang sama.
2. Membandingkan performansi sistem *query by humming* buatan Makarand dan Parag (2018) dengan penggunaan teknik reduksi derau *Fourier Series Decomposition* dan *Spectral Subtraction* pada sistem yang sama.
3. Membandingkan performansi sistem *query by humming* buatan Makarand dan Parag (2018) dengan penggunaan teknik ekstraksi melodi berbasis DNN-LSTM dan teknik reduksi derau *Fourier Series Decomposition* dan *Spectral Subtraction* pada sistem yang sama.

I.4 Batasan Permasalahan

Batasan permasalahan dari penelitian ini adalah:

1. Penelitian ini menggunakan file audio dari forum MIREX[1] sebagai *query* dan basis data.
2. Derau yang dapat ditangani adalah derau yang berjenis *white noise*.

I.5 Metodologi Penelitian

Pada penelitian ini, dilakukan tahapan-tahapan penelitian sebagai berikut:

1. Studi Literatur

   Pada tahapan ini dilakukan studi literatur seputar *query by humming*, *unified algorithm*, *mode normalized frequency, Fourier series decomposition*, *spectral subtraction*, dan teknik ekstraksi melodi berbasis DNN-LSTM.
2. Pembangunan *base-line system query by humming*

   Pada tahapan ini dibangun sistem dasar yang akan digunakan sebagai pembanding dengan sistem yang akan dibangun dalam penelitian ini. Sistem dasar ini dibuat berdasarkan sistem *query by humming* yang diperkenalkan oleh Makarand dan Parag (2018).
3. Eksperimen

[1] https://www.music-ir.org/mirex/wiki/2019:Query_by_Singing/Humming

Pada tahapan ini dilakukan beberapa eksperimen yang akan dijelaskan lebih detail pada bab III. Setiap kombinasi akan diuji coba dan dikomparasi untuk memberikan hasil yang optimum.

4. Evaluasi

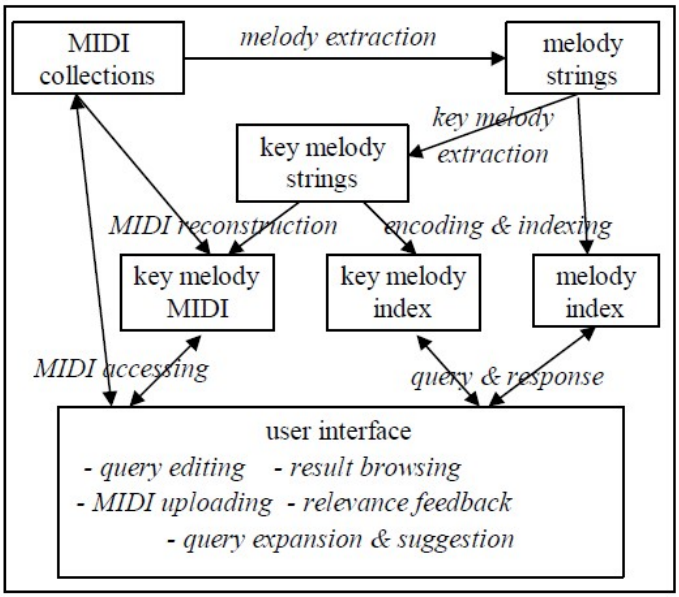
   Pada tahapan evaluasi akan dicari apa kelebihan dan kekurangan dari sistem yang dibangun, serta ditunjukkan poin-poin yang dapat dikembangkan untuk penelitian selanjutnya.

Bab II Tinjauan Pustaka

II.1 *Content-Based Music Retrieval*

Menurut Makarand dan Sahasrabuddhe (2014), sistem temu balik informasi (STBI) untuk domain musik dapat dilakukan dengan dua cara, yaitu berdasarkan metadata dan berdasarkan konten. Metadata untuk musik bisa terdiri dari judul lagu, album, genre, dan artis. Pencarian musik dengan menggunakan metadata akan baik jika metadata untuk seluruh lagu yang ada tersedia lengkap dan setiap lagu memiliki metadata yang tepat. Pencarian musik berdasarkan konten bisa dilakukan dengan melakukan perbandingan pada parameter musik seperti melodi, ritme, dan tempo. Pendekatan ini mengekstrak fitur musik dengan teknik pemrosesan sinyal.

Salah satu teknik pencarian musik berbasis konten diperkenalkan oleh Tseng (1999). Motivasi dari penelitian ini adalah karena banyaknya kebutuhan pencarian musik berdasarkan isi atau konten dari musik itu sendiri dan bukan hanya mengandalkan metadata dari musik. Tseng (1999) menggunakan teknik ekstraksi melodi kunci *(key melody extraction)* untuk mendapatkan melodi yang representatif dan mudah diingat untuk setiap musik dalam basis data. Arsitektur lengkap untuk sistem ini dapat dilihat pada Gambar II.1.

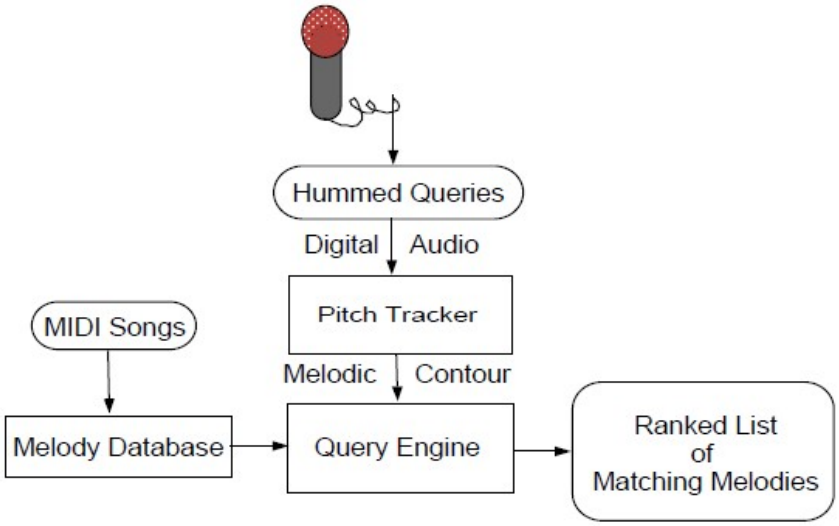


Gambar II.1 Arsitektur sistem *content-based music retrieval* (Tseng, 1999)

Melodi kunci adalah sebuah bagian dari musik yang dapat menggambarkan keseluruhan musik karena merupakan tema atau bagian yang mudah diingat oleh manusia. Ekstraksi dari melodi kunci ini penting untuk dilakukan supaya lebih mudah untuk dicocokkan dengan *query* dari pengguna. Selain itu, proses ini juga dapat mengurangi waktu pencocokan karena membutuhkan komputasi yang lebih ringan. Salah satu teknik ekstraksi melodi kunci ini adalah dengan mengambil pola yang sering muncul dari sebuah musik karena pengulangan adalah aturan yang umum digunakan ketika seseorang membuat musik. Bagian yang berulang ini pula yang umumnya mudah diingat oleh pendengar musik dan akhirnya digunakan sebagai *query* ketika ingin mencari musik tersebut.

II.2 *Query By Humming*

Pada penelitian yang dilakukan oleh Tseng (1999), *query* yang digunakan sebagai bahan pencarian musik adalah rangkaian nada sederhana yang nantinya dicocokkan dengan kumpulan musik yang ada di basis data. Ghias dkk. (1995) menyatakan bahwa cara paling efektif dan natural untuk melakukan pencarian musik adalah dengan menggumamkan lagu yang ingin dicari atau disebut juga dengan *query by humming*. Sistem yang diperkenalkan oleh Ghias dkk. menerima masukan berupa gumaman melalui mikrofon yang kemudian dilakukan ekstraksi nadanya lalu dilakukan pencarian pada basis data dengan *query engine.* Arsitektur sistem ini dapat dilihat pada Gambar II.2.



Gambar II.2 Arsitektur sistem *query by humming* (Ghias dkk., 1995)

Khan dan Mushtaq (2011) menjelaskan empat tahapan dalam proses pencarian musik dengan *query by humming* adalah sebagai berikut:

1. Tahap ekstraksi: fitur dari musik diekstraksi dari keseluruhan file audio. Fitur adalah nilai-nilai sampel dari interval yang sama, bisa berupa nilai nada atau amplitudo
2. Tahap filtrasi: filtrasi derau (*noise*) dari fitur yang sudah diekstraksi dengan membuang sampel yang berada di bawah ambang batas derau
3. Tahap penyimpanan: fitur yang sudah diekstraksi dan dibuang deraunya lalu disimpan dalam basis data
4. Tahap pencocokan: fitur yang sudah disimpan di basis data dicocokkan dengan fitur dari suara gumaman yang menjadi *query* untuk dicari fitur yang paling mirip

Makarand dan Parag (2018) menjelaskan ada tahap tambahan setelah tahap ekstraksi yaitu tahap pembersihan data. Di tahap ini, beberapa teknik pemrosesan sinyal diterapkan untuk mempersiapkan *query* sebelum masuk tahap berikutnya, seperti penghapusan kebisingan di latar belakang, segmentasi melodi, dan lain-lain. Tahap ini penting dilakukan untuk menambah akurasi dari proses pencarian.

Khan dan Mushtaq (2011) juga menjelaskan pembagian *query by humming* menjadi empat kategori berdasarkan teknik pencocokan yang digunakan, yaitu *string matching, tree based search technique, dynamic time warping technique,* dan *Hidden Markov Model (HMM).* Selain dari keempat teknik tersebut, ada tiga teknik lagi yang dapat digunakan sebagai teknik pencocokan, yaitu *location sensitive hash algorithm*, *linear scaling*, dan *unified algorithm*. Masing-masing teknik tersebut akan dijelaskan pada beberapa subbab berikut.

II.2.1 *String Matching*

Teknik *string matching* adalah salah satu teknik paling awal yang digunakan untuk proses pencarian musik dengan *query by humming.* Cara kerja teknik ini adalah dengan mengekstrak representasi melodi dari sebuah musik yang kemudian dikonversi menjadi sinyal audio lalu disusun menjadi representasi kontur yang

sesuai. Ghias dkk. (1995) adalah salah satu kontributor untuk teknik ini dengan memperkenalkan prosedur pelacakan nada (*pitch tracking*). Ghias dkk. (1995) menggunakan tiga buah simbol, yaitu U (*up*), D (*down*), dan S (*same*). Simbol U digunakan ketika nada saat ini lebih tinggi dibandingkan nada sebelumnya, simbol D digunakan ketika nada saat ini lebih rendah dibandingkan nada sebelumnya, dan simbol S digunakan ketika nada saat ini sama dengan nada sebelumnya. Rangkaian pola naik turun nada ini kemudian disusun dan dibandingkan dengan pola yang sudah disimpan pada basis data dengan teknik *brute force.* Tao dkk. (2009) memperkenalkan teknik *Harmonic Product Spectral* untuk melakukan ekstraksi fitur nada dari *query* dan melakukan perbandingan dengan teknik *Approximate String Matching.* Tao dkk. (2009) tidak menyimpan data MIDI sebagai pembanding namun menyimpan suara musik yang sudah digumamkan oleh seseorang dalam basis data.

II.2.2 *Tree Based Search Technique*

Blackburn dan DeRoure (1998) memperkenalkan teknik pencocokan dengan menggunakan struktur *tree* untuk menghemat waktu pencarian. Pada teknik *string matching* biasa, pencocokan dilakukan dengan cara *brute force* ke seluruh data dalam basis data. Dengan struktur *tree,* pencocokan ini bisa dilakukan dengan lebih efisien karena data yang harus dibandingkan lebih sedikit.

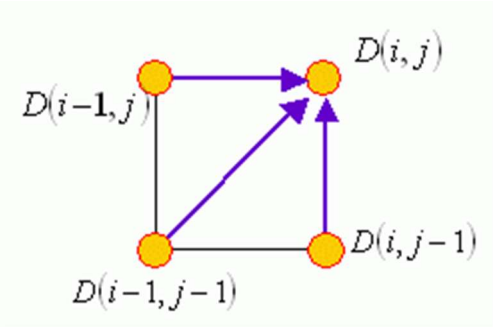
II.2.3 *Dynamic Time Warping Technique*

Teknik *dynamic time warping* banyak digunakan untuk mengukur kesamaan antara dua urutan temporal pada data *time series*. Teknik ini cocok digunakan dalam proses pencarian dengan *query by humming* karena tidak memperhatikan cepat atau lambatnya suara gumaman yang menjadi *query* namun hanya memperhatikan polanya. Koirala dkk. (2018) menggunakan salah satu pengembangan dari teknik ini yaitu *Fast Dynamic Time Warping (FDTW)* untuk membandingkan nada *query* dengan nada musik dalam basis data.

Zhou dkk. (2017) menjelaskan cara kerja *dynamic time warping* adalah dengan mengukur kesamaan antara dua buah *time series,* yaitu rangkaian fragmen *humming*

dan rangkaian template melodi pada basis data. Cara mengukur kesamaan antara dua data tersebut adalah dengan menghitung jarak dari rute paling optimal antara keduanya. Formula untuk menghitung jarak antara dua rangkaian dengan algoritma DTW adalah sebagai berikut:

$$\boldsymbol{g(i,j) = \min \begin{cases} g(i-1,j) + d(i,j) \\ g(i-1,j-1) + d(i,j) \\ g(i,j-1) + d(i,j) \end{cases}} \quad \text{(II.1)}$$



Gambar II.3 Poin dalam algoritma *dynamic time warping*[2]

Formula II.1 di atas menghitung rute terpendek antara dua rangkaian dari tiga rute yang ada. Dalam formula tersebut, *i* adalah sebuah poin dalam sumbu X, *j* adalah sebuah poin dalam sumbu Y, *g(i,j)* adalah salah satu dari tiga kemungkinan posisi poin sebelumnya, dan *d(i,j)* menunjukkan jarak Euclidean antara X dan Y pada poin *(i,j)*. Contoh visual dari tiga kemungkinan rute yang bisa diambil dapat dilihat pada Gambar II.3, di mana *D(i,j)* menunjukkan poin yang menjadi tujuan dan tiga poin lain adalah poin sebelumnya.

Dalam sistem yang diperkenalkan oleh Zhou dkk. (2017), rangkaian dua dimensi berisi nada relatif dan durasi relatif dari fragmen *humming* adalah sumbu X sementara rangkaian melodi dalam basis data yang akan dibandingkan adalah sumbu Y. Dengan algoritma DTW, semua rangkaian melodi dalam basis data akan ditelusuri dan dihitung jaraknya dengan fragmen *humming* yang menjadi *query* lalu diurutkan. Semakin kecil jaraknya maka nilai kesamaan antara fragmen *humming*

[2] http://mirlab.org/jang/books/dcpr/dpDtw.asp?title=8-4%20Dynamic%20Time%20Warping

dengan melodi tersebut makin tinggi. Sistem akan mengambil beberapa melodi dengan kesamaan tertinggi sebagai hasil.

Menurut Zhou dkk. (2017), algoritma DTW lebih efektif untuk menyelesaikan beberapa error ketika melakukan pencocokan melodi dan nilai akurasinya cenderung tinggi. Namun, dalam proses mencari rute optimal, algoritma DTW harus melakukan perhitungan yang cukup kompleks sehingga dibutuhkan komputasi yang tinggi. Kondisi ini membuat waktu proses yang dibutuhkan cukup tinggi.

II.2.4 *Hidden Markov Model* (HMM)

Hidden Markov Model (HMM) pada dasarnya adalah model statistik yang memeriksa semua opsi yang memungkinkan dalam melakukan pencocokan antara *query* dengan basis data. Shih dkk. (2003) memperkenalkan sistem *query by humming* dengan HMM. Ekstraksi fitur dilakukan dengan menggunakan *Mel-Frequency Cepstral Coefficients* (MFCC) yang biasa digunakan pada pengenalan ucapan. Tinggi rendah nada diukur dengan *Gaussian Mixture Models* (GMMs). GMM adalah salah satu teknik pengukuran paling stabil yang digunakan untuk pengelompokkan (*clustering*).

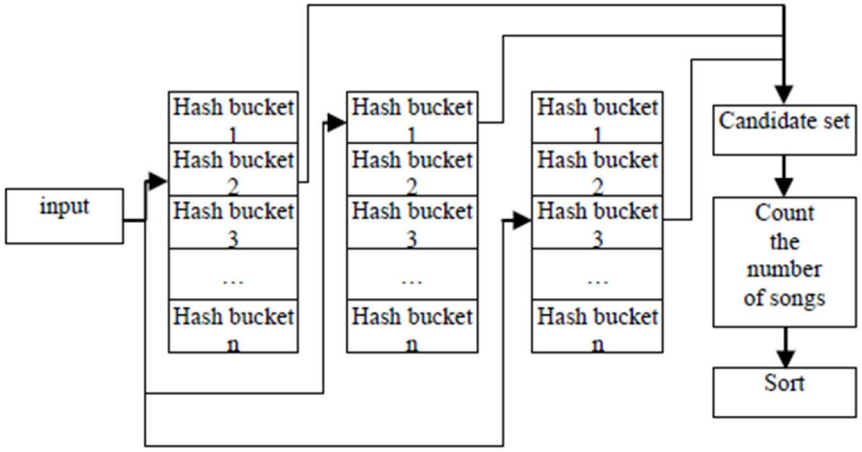
Cara kerja dari HMM untuk proses *query by humming* ini adalah dengan menghitung probabilitas kemunculan sebuah rangkaian melodi yang merupakan *hidden state,* jika diberikan sebuah rangkaian fragmen *humming* sebagai *query* yang merupakan *observable state.* Shih dkk. (2003) menggunakan program HTKEdit yang ditulis berdasarkan *Hidden Markov Model Toolkit (HTK)*[3].

II.2.5 *Location Sensitive Hash Algorithm*

Algoritma *Location Sensitive Hash* (LSH) adalah salah satu algoritma pencocokan melodi yang digunakan oleh Zhou dkk. (2017) dalam penelitiannya. Prinsip kerja dari algoritma ini adalah memetakan data masukan dengan menggunakan fungsi hash sehingga data dipetakan ke dalam *hash bucket* berbeda pada tabel hash. Data

[3] "Hidden Markov Model Toolkit," URL: http://htk.eng.cam.ac.uk/

yang mirip pada data masukan mempunyai probabilitas lebih tinggi untuk dipetakan ke dalam *bucket* yang sama dan kecil kemungkinan untuk data berbeda dipetakan ke dalam *bucket* yang sama. Zhou dkk. (2017) menggunakan tiga buah fungsi hash untuk melakukan pemetaan rangkaian nada relatif. Diagram cara kerja algoritma ini dapat dilihat pada Gambar II.4 berikut.



Gambar II.4 Diagram cara kerja LSH (Zhou dkk., 2017)

Pertama-tama, data musik dalam basis data diproses untuk dihasilkan vektor untuk fitur nadanya dalam bentuk String. Setiap musik memiliki label berupa judul lagunya. Setiap String yang berisi fitur nada lalu diproses dengan fungsi hash sehingga dihasilkan pemetaan setiap String ke dalam *hash bucket* berbeda pada tabel hash. Setelah proses dengan tiga fungsi hash selesai, akan dihasilkan tiga buah tabel hash dengan label judul lagunya. Setiap tabel memiliki nilai hash yang berhubungan dengan sebuah lagu dalam basis data. Fragmen *humming* yang sudah dilakukan ekstraksi fitur nadanya diproses dengan tiga fungsi hash lalu semua data dalam *hash bucket* yang terpetakan oleh fragmen *humming* diambil.

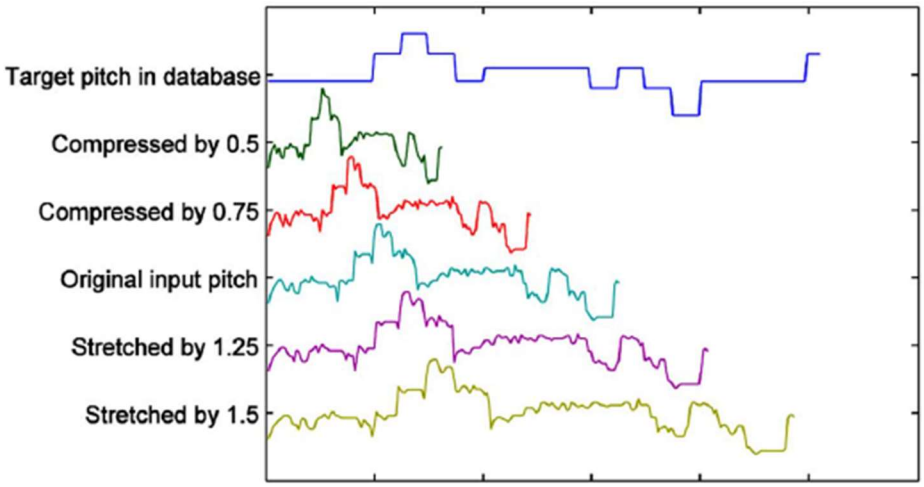
Dalam contoh ilustrasi pada Gambar II.4, fragmen *humming* terpetakan ke *bucket* 2 pada tabel 1 oleh fungsi pertama, terpetakan ke *bucket* 1 pada tabel 2 oleh fungsi kedua, dan terpetakan ke *bucket* 3 pada tabel 3 oleh fungsi ketiga. Kemudian, semua data pada *bucket* 2 di tabel 1, *bucket* 1 di tabel 2, dan *bucket* 3 di tabel 3 diambil. Data ini merujuk ke sebuah lagu yang sedang dicari, dan semakin banyak data

merujuk ke satu lagu yang sama, semakin tinggi pula ranking lagu tersebut dalam hasil pencarian.

II.2.6 *Linear Scaling*

Linear scaling adalah salah satu teknik pencocokan, selain Dynamic Time Warping (DTW), untuk menangani masalah tempo yang berbeda antara *query* dan musik dalam basis data menurut Wang dan Jang (2015). Prinsip kerja dari *linear scaling* adalah dengan melakukan kompresi dan peregangan pada vektor nada yang menjadi *query* dengan harapan salah satu hasil dari kompresi atau peregangan tersebut cocok dengan vektor nada yang menjadi target dalam basis data.

Semisal vektor masukan mempunyai durasi *d* detik, vektor tersebut akan dikompresi dan diregangkan sehingga mendapat sebanyak *r* buah versi dengan interval durasi sama antara s_{min} *x d* dan s_{max} *x d*, di mana s_{min} (<1) dan s_{max} (>1) adalah faktor skala minimal dan maksimal. Gambar II.5 berikut adalah contoh hasil proses *linear scaling* dengan nilai *r* adalah 5, s_{min} adalah 0.5, dan s_{max} adalah 1.5. Dari contoh di bawah, hasil terbaik didapat ketika faktor skala yang digunakan adalah 1.25.



Gambar II.5 Contoh proses *linear scaling* (Wang dan Jang, 2015)

Penelitian yang dilakukan oleh Wang dan Jang (2015) menghasilkan kesimpulan bahwa dengan menambah nilai *r* akan menghasilkan tingkat akurasi yang lebih tinggi, namun membutuhkan waktu komputasi yang lebih lama.

II.2.7 *Unified Algorithm*

Unified algorithm merupakan penggabungan dari beberapa algoritma pencocokan yang bekerja bergantung pada kebutuhan. Jika hasil yang diinginkan tidak didapatkan pada algoritma yang sedang digunakan, maka *unified algorithm* ini akan menggunakan algoritma selanjutnya. Ada empat algoritma yang digunakan dengan urutan penggunaannya sebagai berikut: terdiri dari tiga buah algoritma yang menggunakan struktur indeks terbalik (*inverted index structure*) dengan pencocokan pola n-gram, yaitu *relative pitch 4-grams* (RP4G)*, 3-grams* (RP3G)*,* dan *2-grams* (RP2G), serta sebuah algoritma untuk melakukan normalisasi frekuensi yang disebut *mode normalized frequency* (MNF) dengan pencocokan pola menggunakan metode *edit distance*.

Struktur indeks terbalik digunakan dengan nada relatif n-gram yang sudah dikalkulasi sebelumnya. Sebagai contoh, jika nada yang didapatkan dari *query* adalah do re mi si dan jika dikonversi ke dalam not angka menjadi 1 2 3 7, maka bisa dihitung jarak relatif antar nadanya adalah +1 +1 +4. Dari jarak relatif tersebut, jika diambil 2-gramnya adalah +1 dan +4. Indeks terbalik untuk string +1 akan berisi seluruh musik dalam basis data yang mengandung string +1 dengan terurut berdasarkan jumlah kemunculan dari string +1. Untuk setiap musik dalam basis data, akan dihitung jumlah kemunculan dari setiap kemungkinan 2-gram dari *query*. Semakin banyak jumlah kemunculan dari 2-gram pada sebuah musik maka kemiripannya semakin tinggi dengan *query*. Dari contoh di atas, jika diambil 3-gramnya adalah +1 +1 dan +1 +4 dan untuk 4-gramnya adalah +1 +1 +4. Proses penghitungan yang dilakukan untuk 3-gram dan 4-gram sama persis dengan 2-gram.

Mode normalized frequency (MNF) adalah algoritma yang digunakan untuk mengkonversi string nada relatif menjadi representasi dalam huruf yang relatif kepada nada yang paling sering muncul (modus). Nada yang paling sering muncul akan direpresentasikan menjadi huruf N yang merupakan alfabet tengah. Nada relatif lain akan direpresentasikan relatif terhadap nada tersebut, sehingga nada dengan jarak +2 akan ditulis menjadi P dan -2 menjadi L. Algoritma untuk

melakukan pemrosesan *query* dengan normalisasi frekuensi menggunakan modus ini dapat dijabarkan sebagai berikut:

1. Baca daftar nada dari *query*
2. Cari nada dengan frekuensi kemunculan paling tinggi (N) dari daftar nada
3. Cari nada lain dan frekuensinya
4. Cari rentang antar nada untuk mencakup seluruh rentang frekuensi
5. Konversi daftar nada dari *query* ke representasinya
6. Baca hasil konversi nada
7. Jika nada muncul 10 kali berturut turut, masukkan ke daftar keluaran
8. Lanjutkan membaca sampai akhir
9. Keluarkan hasil normalisasi

Berikut adalah contoh masukan dan keluaran dari algoritma MNF, dengan tanda bintang (*) di setelah not nada berarti nada tersebut berada satu oktaf di atas dan nada re* sebagai nada yang paling sering muncul:
Pola rangkaian nada: do re* do* re* fa* do* la do* re* la sol la do*
Pola nada relatif: +8 −1 +1 +2 −3 −2 +2 +1 −3 −1 +1 +2
Pola *query* hasil normalisasi: FNMNPMKMNKJKM

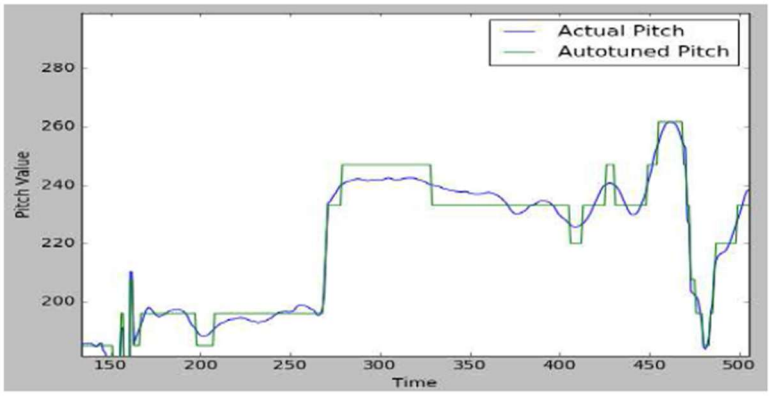
Unified algorithm memperkecil kemungkinan hasil dengan pendekatan n-gram lalu algoritma MNF diaplikasikan untuk mengukur kesamaan pola menggunakan *edit distance* pada daftar hasil yang sudah lebih kecil. *Edit distance* adalah metode pencocokan pola yang menggunakan pengukuran jarak Euclidian dengan jarak adalah angka minimum yang dibutuhkan untuk menambah, menghapus, atau mengganti sebuah string supaya sama dengan string lain. Semakin kecil nilai jarak yang didapat maka semakin mirip *query* dengan musik di basis data. MNF menghilangkan kebutuhan untuk *query transpose* yang pada penelitian lain dibutuhkan untuk mengatasi masalah berupa nada dasar yang berbeda pada *query* yang sama.

II.3 Filtrasi Derau

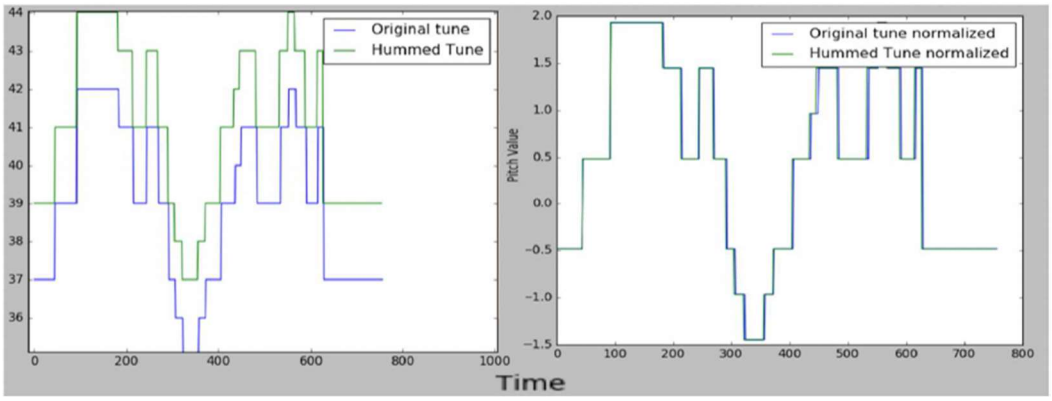
Filtrasi derau adalah tahapan pertama dalam sistem pencarian musik dengan *query by humming* yang dilakukan sebelum tahap ekstraksi melodi. Proses ini dilakukan dengan menerapkan beberapa teknik pemrosesan sinyal untuk mempersiapkan *query* sebelum masuk tahap berikutnya. Hasil keluaran yang diharapkan dari proses ini adalah sinyal suara yang lebih sedikit mengandung derau supaya dapat meningkatkan hasil akurasi dari pencarian musik dengan *query by humming*. Salah satu penelitian yang sudah menerapkan tahapan ini dalam sistem *query by humming* adalah Koirala dkk. (2018) yang menerapkan teknik *auto tuning* dan *pitch normalization*. Selain itu, ada Siam dkk. (2019) yang memperkenalkan salah satu teknik untuk melakukan peningkatan kualitas sinyal suara dengan *Fourier series decomposition* dan *spectral subtraction* untuk domain identifikasi pembicara namun belum pernah dicoba untuk sistem *query by humming*.

II.3.1 *Auto Tuning* dan *Pitch Normalization*

Salah satu teknik untuk menambah akurasi pencarian musik dengan *query by humming* adalah dengan melakukan *auto tuning* dan *pitch normalization.* Tujuan dari kedua proses tersebut adalah menghasilkan *query* dan musik dengan nada yang tepat. Koirala dkk. (2018) melakukan penelitian untuk melihat efek dari proses tersebut terhadap performa pencarian musik dengan *query by humming.* Gambar II.6 dan Gambar II.7 berikut menjelaskan proses *auto tuning* dan *pitch normalization* yang dilakukan oleh Koirala dkk.



Gambar II.6 Proses *auto-tuning* (Koirala dkk., 2018)



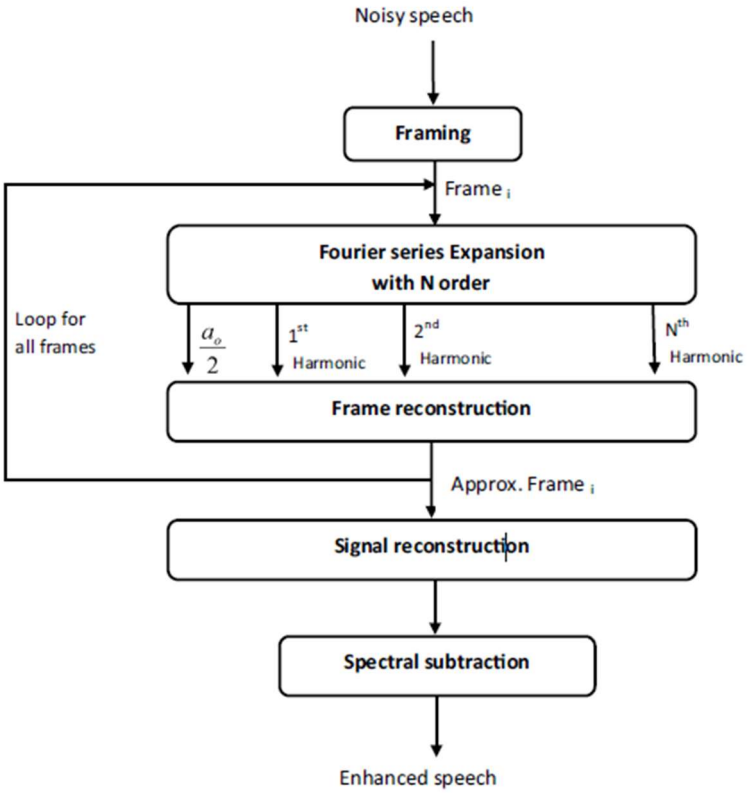
Gambar II.7 Proses *pitch normalization* (Koirala dkk., 2018)

Proses *auto tuning* pada *query* dan musik dalam sistem *query by humming* menambah akurasi dari sistem secara drastis. Sementara, proses *pitch normalization* tidak memberikan efek yang besar terhadap akurasi sistem jika jumlah musik yang disimpan dalam basis data relatif sedikit. Proses ini baru bisa memberikan efek yang baik ketika jumlah musik yang disimpan di basis data ditambah.

II.3.2 *Fourier Series Decomposition* dan *Spectral Subtraction*

Salah satu teknik untuk melakukan peningkatan kualitas suara yang pernah dicoba untuk domain identifikasi pembicara adalah dengan *Fourier series decomposition* dan *spectral subtraction* yang diperkenalkan oleh Siam dkk. (2019). Cara kerja dari teknik ini adalah pertama-tama sinyal suara masukan disegmentasi menjadi beberapa *frame* kecil. Kemudian, setiap frame didekomposisi menjadi N buah harmonik menggunakan deret *Fourier*. Lalu, setiap *frame* direkonstruksi kembali dengan menjumlahkan beberapa harmonik untuk mendapatkan estimasi *frame* akhir yang harapannya memiliki derau lebih rendah daripada yang asli. Proses ini diulang sampai semua *frame* selesai diproses. Setelah itu, *spectral subtraction* diterapkan pada sinyal yang sudah direkonstruksi untuk memperoleh hasil akhir berupa sinyal suara yang sudah ditingkatkan kualitasnya. Alasan dilakukan *framing* sinyal suara sebelum ekspansi Fourier adalah supaya didapatkan hasil sinyal yang lebih detail

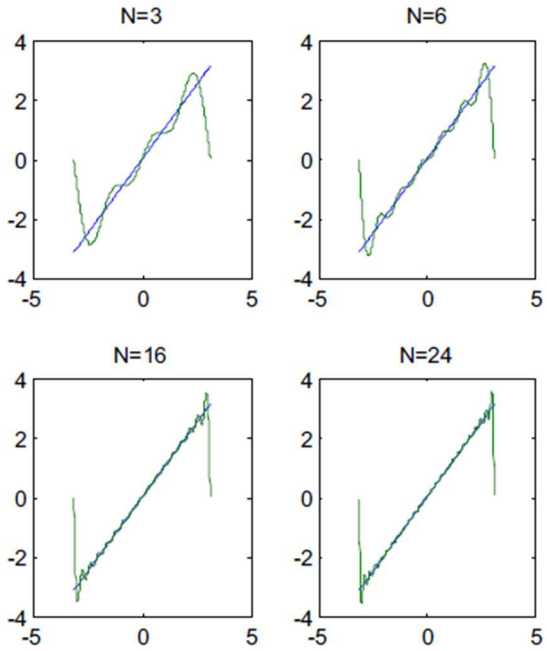
karena semakin kecil skala sinyal maka semakin detail hasil yang didapat dari ekspansi Fourier. Ilustrasi dari proses ini dapat dilihat pada Gambar II.8 berikut.



Gambar II.8 Alur peningkatan kualitas suara (Siam dkk., 2019)

II.3.2.1 Ekspansi Deret Fourier

Deret Fourier (*Fourier series*) menguraikan sinyal ke dalam sejumlah (kemungkinan tak hingga) fungsi harmonik sederhana yang disebut sinus dan kosinus. Harmonik ini mempunyai amplitudo dan frekuensi yang menutupi hampir seluruh spektrum, di mana frekuensi dari suatu harmonik lebih tinggi daripada frekuensi harmonik sebelumnya. Hanya sejumlah kecil harmonik ini yang bisa mengestimasi sinyal dengan kemungkinan beberapa distorsi. Semakin tinggi jumlah harmonik yang diambil untuk menggambarkan sinyal, semakin rendah jumlah distorsi yang muncul di sinyal, seperti dapat dilihat pada Gambar II.9. Jumlah harmonik yang diambil untuk melakukan aproksimasi sinyal disebut orde atau *N* dari deret Fourier.



Gambar II.9 Aproksimasi sinyal dengan deret Fourier pada jumlah harmonik yang berbeda (Siam dkk., 2019)

Deret Fourier dari sebuah sinyal diskrit *y(k)* adalah sebagai berikut:

$$\boldsymbol{y} \approx \frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{a_0} + \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \boldsymbol{a_m} \boldsymbol{\cos}\left(\frac{\boldsymbol{2\pi m}}{\boldsymbol{L}}\boldsymbol{y}\right) + \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \boldsymbol{b_m} \boldsymbol{\sin}\left(\frac{\boldsymbol{2\pi m}}{\boldsymbol{L}}\boldsymbol{y}\right) \quad \text{(II.2)}$$

di mana *M* adalah orde dari deret Fourier, $1 \leq M < \infty$, *L* adalah panjang dari sinyal, dan

$$\boldsymbol{a_0} = \frac{\boldsymbol{1}}{\boldsymbol{L}}\sum_{\boldsymbol{k=1}}^{\boldsymbol{L}} \boldsymbol{y(k)} \quad \text{(II.3)}$$

$$\boldsymbol{a_m} = \frac{\boldsymbol{2}}{\boldsymbol{L}}\sum_{\boldsymbol{k=1}}^{\boldsymbol{L}} \boldsymbol{y(k)} \boldsymbol{\cos}\left(\frac{\boldsymbol{2\pi mk}}{\boldsymbol{L}}\right) \quad \text{(II.4)}$$

$$\boldsymbol{b_m} = \frac{\boldsymbol{2}}{\boldsymbol{L}}\sum_{\boldsymbol{k=1}}^{\boldsymbol{L}} \boldsymbol{y(k)} \boldsymbol{\sin}\left(\frac{\boldsymbol{2\pi mk}}{\boldsymbol{L}}\right) \quad \text{(II.5)}$$

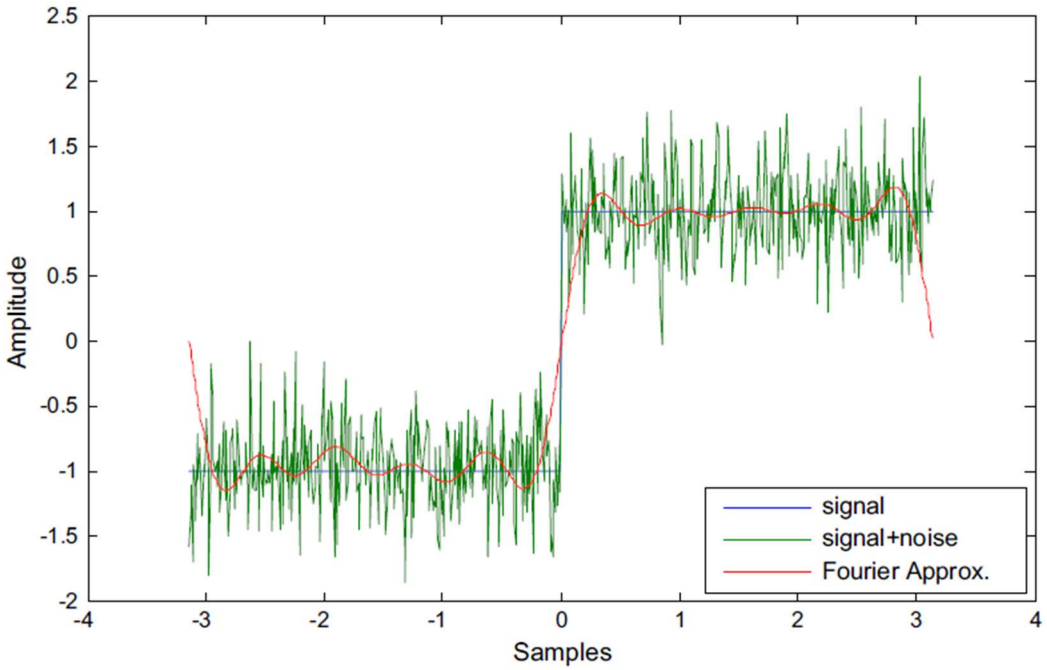
Rumus $a_m \cos\left(\frac{2\pi m}{L}y\right) + b_m \sin\left(\frac{2\pi}{L}y\right)$ pada persamaan II.2 disebut dengan harmonik ke-*m* pada deret Fourier, sehingga persamaan

$a_1 \cos\left(\frac{2\pi(1)}{L}y\right) + b_1 \sin\left(\frac{2\pi(1)}{L}y\right)$ disebut dengan harmonik ke-1,

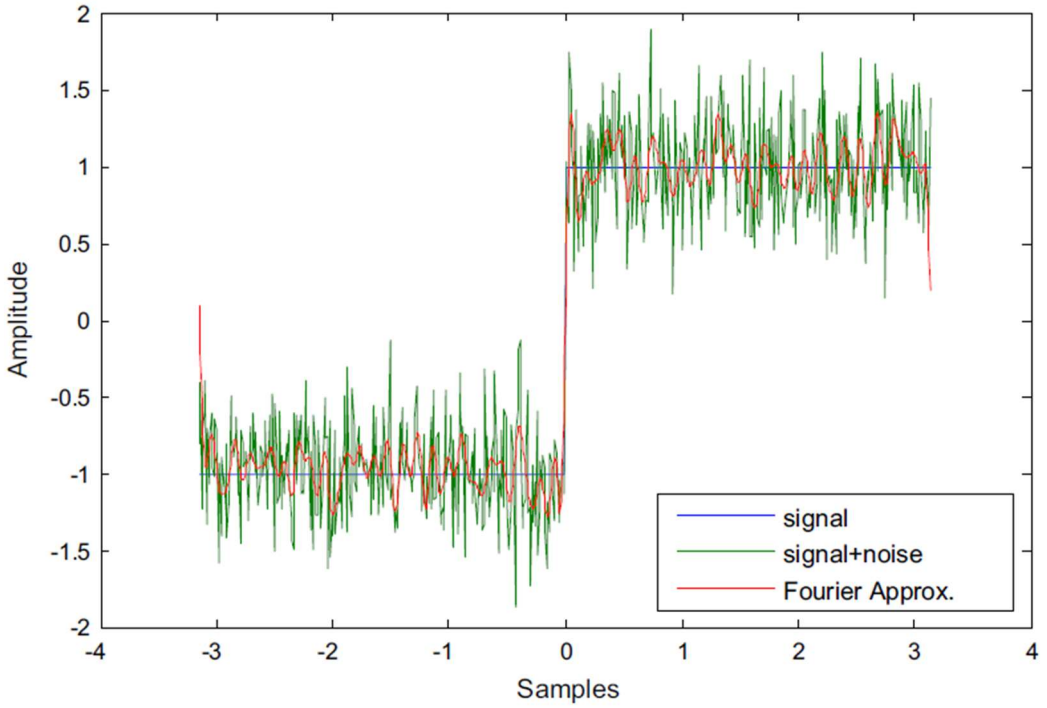
$a_2 \cos\left(\frac{2\pi(2)}{L}y\right) + b_2 \sin\left(\frac{2\pi(2)}{L}y\right)$ disebut dengan harmonik ke-2, dan seterusnya.

Deret Fourier menguraikan sinyal menjadi harmonik sederhana (sinus dan kosinus), masing-masing dengan frekuensi tunggal, mencakup seluruh bandwidth sinyal. Setiap harmonik diberi bobot dengan amplitudo (a_n dan b_n) untuk membentuk gelombang dari sinyal. Sinyal dengan frekuensi rendah bisa diekpresikan dengan beberapa harmonik pertama, dan jumlah harmonik bertambah dengan meningkatkan komponen frekuensi dari sinyal. Untuk sinyal bersih yang terkontaminasi *Additive White Gaussian Noise* (AWGN), spektrum diperluas menutupi komponen frekuensi tinggi yang terkandung dalam sinyal derau sedemikian rupa sehingga kekuatan sinyal bersih menempati bagian bawah spektrum sedangkan kekuatan derau mencakup seluruh spektrum.

Ketika mengaplikasikan ekspansi deret Fourier, kita dapat mengambil aproksimasi dari sinyal bersih dengan mengambil hanya beberapa harmonik awal yang dapat menunjukkan sebagian besar sinyal dengan komponen frekuensi derau yang kecil. Harmonik yang lebih tinggi merepresentasikan sebagian besar derau dan beberapa komponen sinyal frekuensi tinggi diabaikan. Gambar II.10 dan Gambar II.11 menunjukkan sinyal bersih, sinyal berderau hasil penambahan AWGN dengan *Signal-to-Noise Ratio* (SNR) = 10 dB, dan aproksimasi Fourier untuk sinyal berderau dengan $N = 10$ dan $N = 60$. Semakin tinggi orde dari aproksimasi Fourier untuk sinyal berderau, semakin banyak juga derau pada gelombang yang dihasilkan.



Gambar II.10 Aproksimasi Fourier untuk sinyal berderau ($N = 10$) (Siam dkk., 2019)



Gambar II.11 Aproksimasi Fourier untuk sinyal berderau ($N = 60$) (Siam dkk., 2019)

II.3.2.2 *Spectral Subtraction*

Estimasi dari spektrum sinyal bersih bisa didapat dengan mengurangkan estimasi spektrum derau dari spektrum berderau. Estimasi spektrum derau bisa didapat dari periode diam (*silence periods*) dalam sinyal suara, yang hanya mengandung derau latar belakang (*background noise*) yang ada sepanjang awal sampai akhir dalam rekaman suara. Semisal,

$$\boldsymbol{o(n) = s(n) + v(n)} \tag{II.6}$$

di mana $o(n)$ adalah sinyal suara yang belum bersih, yang merupakan kombinasi dari derau $v(n)$ dan sinyal bersih $s(n)$. Jika *Fast Fourier Transform* (FFT) adalah

$$\boldsymbol{O(\omega) = S(\omega) + V(\omega)} \tag{II.7}$$

maka $S(\omega)$ dapat ditulis sebagai

$$\boldsymbol{S(\omega) = O(\omega) - V(\omega)} \tag{II.8}$$

$$\boldsymbol{S(\omega) = |O(\omega)|e^{j\theta_o} - |V(\omega)|e^{j\theta_v}} \tag{II.9}$$

diasumsikan fase sinyal derau θ_v sama dengan fase sinyal berderau θ_o.

$$\boldsymbol{S(\omega) = [|O(\omega)| - |V(\omega)|]e^{j\theta_o}} \tag{II.10}$$

$$\boldsymbol{\hat{S}(\omega) = [|O(\omega)| - |\mu(\omega)|]e^{j\theta_o}} \tag{II.11}$$

di mana $\hat{S}(\omega)$ adalah estimasi spektrum dari sinyal bersih dan $\mu(\omega) = mean\{|V(\omega)|\}$ adalah nilai rata-rata yang diambil saat periode diam.

Dengan mengambil invers dari FFT, didapatkan estimasi dari sinyal bersih $s(n)$. Performa dari *spectral subtraction* sangat bergantung pada jumlah derau yang berhasil diestimasi. Jika derau yang diestimasi terlalu rendah, derau sisa (*residual noise*) masih akan dapat terdengar, sementara jika derau yang diestimasi terlalu tinggi, beberapa informasi penting akan ikut hilang bersama derau.

II.4 Ekstraksi Melodi

Tahap kedua dalam proses pencarian musik dengan *query by humming* adalah ekstraksi melodi. Menurut Cao dkk. (2020), ekstraksi melodi bertujuan untuk menghasilkan urutan frekuensi nilai yang sesuai dengan nada melodi dominan dari rekaman musik. Ada beberapa teknik yang pernah digunakan dalam penelitian sebelumnya, seperti teknik *pitch tracking* yang digunakan oleh Ghias dkk. (1995),

Blackburn dan DeRoure (1998), dan Yang dkk., (2010), lalu ada teknik *harmonic product spectral* yang digunakan oleh Tao dkk. (2009), dan teknik *autocorrelation function* yang digunakan oleh Zhou dkk. (2017). Ada salah satu teknik ekstraksi melodi baru yang diperkenalkan oleh Cao dkk. (2020) dan belum pernah digunakan dalam sistem *query by humming*, yaitu *multi task learning* berbasis DNN-LSTM. Selain itu, ada juga teknik *mel-frequency cepstral coefficients* (MFCC) yang digunakan oleh Shih dkk. (2003).

II.4.1 *Multi task learning* berbasis DNN-LSTM

Teknik ekstraksi melodi berbasis DNN-LSTM yang diperkenalkan oleh Cao dkk. (2020) ini merupakan komposisi antara dua fungsi, yaitu *pitch estimation* dan *voicing detection*. Pendekatan yang digunakan merupakan pendekatan pembelajaran dengan tugas ganda (*multi-task learning approach*) yang melakukan dua fungsi tersebut secara bersamaan. Estimasi nada (*pitch estimation*) bertugas untuk melakukan estimasi nilai frekuensi nada dari sebuah melodi sedangkan deteksi suara (*voicing detection*) bertugas untuk mengidentifikasi ada atau tidaknya melodi. Keluaran dari fungsi estimasi nada adalah sebuah kelas nada atau not yang berisi nada dalam rentang frekuensi tertentu sedangkan keluaran dari fungsi deteksi suara adalah biner antara ada atau tidaknya melodi. Hasil eksperimen menunjukkan bahwa dengan melakukan pembelajaran untuk kedua fungsi tersebut secara bersamaan dapat meningkatkan akurasi dan mempunyai kemampuan generalisasi yang lebih baik.

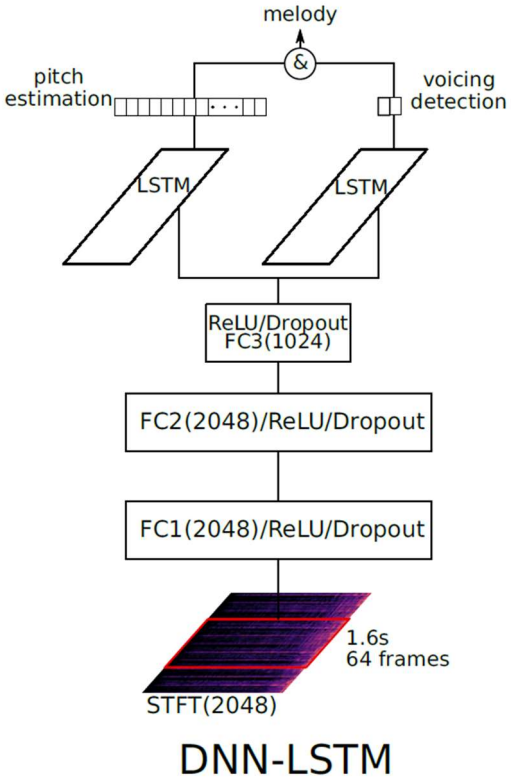
Teknik ini melakukan ekstraksi melodi berdasarkan frame, dengan memprediksi frekuensi instan setiap 23 milidetik. Tugas estimasi nada dianggap seperti tugas klasifikasi biasa. Nada dengan rentang hampir 5 oktaf dengan frekuensi antara 55 Hz sampai 1,76 kHz diambil sehingga model dapat menangkap sebagian besar nada melodi dan menjaga distribusi data pada setiap kelas nada seimbang. Untuk frame melodi yang memiliki suara, algoritma diharapkan mengeluarkan nilai frekuensi yang benar, yang mana dianggap benar jika selisihnya dalam batas 50 sen (setengah nada) dengan nada aslinya. Jika frekuensi asli dinyatakan dengan f Hz, perhitungan kelas nada $Clz(f)$ yang sesuai adalah sebagai berikut:

$$\boldsymbol{Clz(f)} = \left\lfloor \boldsymbol{N_{pitchClass}} \cdot \boldsymbol{\log_{\frac{f_{high}}{f_{low}}}\left(\frac{f}{f_{low}}\right)} \right\rfloor \quad \text{(II.12)}$$

di mana $N_{pitchClas}$ adalah jumlah kelas nada yang ada, dan $[f_{low}, f_{high})$ adalah rentang frekuensi nada yang dipetakan kepada setiap kelas nada pada $[0: N_{pitchClass})$. Frekuensi yang berada di luar batas akan dijadikan nilai maksimum atau minimum.

Tugas deteksi suara adalah menentukan apakah ada atau tidak melodi pada suatu titik yang dianggap sebagai tugas klasifikasi biner. Karena lebih banyak jumlah kelas pada tugas estimasi nada dibandingkan jumlah kelas pada tugas ini, maka tugas estimasi nada dianggap sebagai tugas utama dan deteksi suara dianggap sebagai tugas tambahan.

Arsitektur dari sistem ekstraksi melodi ini menggunakan *deep neural network* (DNN) dan *recurrent neural network* (RNN) dalam bentuk *long short term memory* (LSTM). Arsitektur DNN untuk kedua tugas tersebut menggunakan tiga *hidden layers* dengan jumlah unit 2048, 2048, dan 1024 dengan ReLU sebagai fungsi nonlinear sementara LSTM digunakan untuk masing-masing tugas secara terpisah. Model mengambil beberapa frame dari hasil *Short Time Fourier Transform* (STFT) pada audio masukan sebagai vektor fitur masukan dan menghasilkan keluaran berupa dua label untuk setiap jeda waktu (*timestamp*) yang merupakan hasil prediksi dari estimasi nada dan deteksi suara. Dua label tersebut kemudian disatukan untuk membentuk prediksi melodi pada tahap terakhir. Ilustrasi dari arsitektur tersebut dapat dilihat pada Gambar II.12.



Gambar II.12 Arsitektur teknik ekstraksi melodi berbasis DNN-LSTM (Cao dkk., 2020)

Untuk menangkap dependensi temporal, LSTM mengambil keluaran dari *hidden layer* terakhir dari DNN sebagai masukan yang berisi hasil pembelajaran representasi oleh DNN. Jaringan LSTM adalah jenis khusus dari RNN yang mampu mempelajari dependensi jangka panjang dan menghubungkan informasi sebelumnya dengan tugas yang dikerjakan saat ini. Untuk setiap tugas, *2-layer bi-LSTM* dengan 64 unit dilatih sehingga model akan mempunyai vektor masukan dengan ukuran (64, 2048) yang menangkap $\frac{63x512+204}{22050} \approx 1{,}6$ detik potongan audio dengan *sampling rate* 22050 Hz. Jeda STFT diatur menjadi 2048 frame dan *hop size* 512 frame. Dua *fully connected layer* terakhir akan membuat prediksi untuk masing-masing tugas estimasi nada dan deteksi suara.

Untuk menggabungkan dua hasil prediksi dari ekstraksi melodi, semisal $label_{pe} = Clz(f) \in [0: N_{pitchClass})$ untuk *pitch estimation*, di mana fungsi Clz dari persamaan II.12 memetakan frekuensi ke kelas nada, dan $label_{vd} \in \{0,1\}$ untuk

deteksi suara, di mana 0 berarti tidak ada melodi dan 1 berarti ada melodi, maka melodi hasil ekstraksi didapat dari:

$$\boldsymbol{instantFreq} = \begin{cases} \boldsymbol{Clz^{-1}(label_{pe})}, \textbf{ jika } \boldsymbol{label_{vd}} = \mathbf{1}. \\ \mathbf{0}, \qquad\qquad \textbf{jika } \boldsymbol{label_{vd}} = \mathbf{0}. \end{cases} \quad \textbf{(II.13)}$$

II.4.2 *Mel-Frequency Cepstral Coefficients (MFCC)*

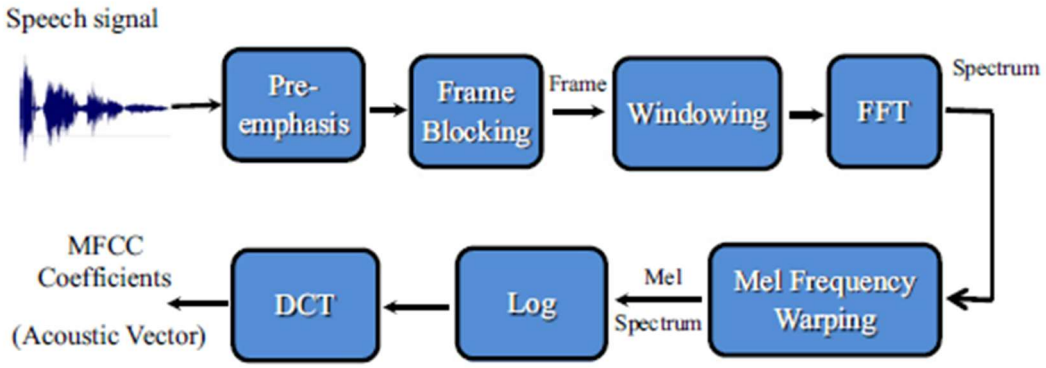
Mel-Frequency Cepstral Coefficients (MFCC) adalah salah satu teknik yang sering digunakan untuk melakukan ekstraksi fitur dari sinyal suara dalam domain pengenalan ucapan. Menurut Shih dkk. (2003), karena produksi suara *humming* manusia mirip dengan produksi ucapan, fitur yang digunakan untuk membedakan fonem pada *Automatic Speech Recognition (ASR)* dapat juga digunakan untuk memodelkan nada pada pengenalan *humming.* Menurut Siam dkk. (2019), beberapa fitur penting yang dapat diekstrak dengan MFCC adalah frekuensi, *loudness,* energy, dan spectrum.

Menurut Kurzekar dkk. (2014), MFCC didapatkan dari analisis *nonlinear filterbank* yang diadaptasi dari mekanisme pendengaran manusia sehingga umum digunakan untuk sistem pengenalan ucapan. MFCC digunakan untuk mengkarakterisasi bentuk akustik dari nada *humming.* Pada penelitian Shih dkk. (2003), dipilih 26 *filterbank channels* dan 12 MFCC yang pertama dipilih sebagai fitur. Selain itu, energi adalah fitur penting dalam pengenalan *humming* untuk melakukan segmentasi antar nada. Pada umumnya, variasi energi yang berbeda akan terjadi selama transisi dari satu nada ke nada lainnya. Efek ini terutama ditingkatkan karena pengguna diminta untuk bersenandung *(humming)* menggunakan suara dasar yang merupakan kombinasi dari konsonan dan vokal (semisal “da” dan “la”).

MFCCs bekerja seperti sistem persepsi pendengaran manusia, yang tidak dapat merasakan frekuensi lebih tinggi dari 1 kHz, secara linear. Dengan demikian, ekstraksi MFCC membutuhkan dua jenis filter yang ditempatkan secara linear pada frekuensi rendah di bawah 1 kHz dan secara logaritmik di atas 1 kHz. Output dari filter ini selaras dengan skala Mel yang bisa dijelaskan pada formula berikut, di mana *Mel* adalah frekuensi Mel dan *f* adalah frekuensi linier dalam Hz.

$$\boldsymbol{Mel(f) = 2595 * log_{10}\left(1 + \frac{f}{700}\right)} \quad \text{(II.14)}$$

Alur proses lengkap ekstraksi fitur dengan MFCC dapat dilihat pada Gambar II.13 berikut. Operasi proses ekstraksi fitur dengan MFCC dimulai dengan menangkap sinyal ucapan input melalui mikrofon dengan frekuensi sampling $F_s \geq 8$ KHz. Ini dilakukan untuk memastikan sebagian besar energi terkandung dalam sinyal dengan frekuensi $300 \leq F_m \leq 3400$ Hz dapat ditangkap. Kemudian sinyal sampel melewati tujuh langkah komputasi sampai didapatkan MFCCs (cetak suara) dari langkah terakhir.



Gambar II.13 Diagram proses ekstraksi fitur dengan MFCC (Siam dkk., 2019)

II.5 *Mean Reciprocal Rank (MRR)*

Salah satu metriks untuk mengukur tingkat akurasi dari sebuah sistem temu balik informasi (STBI) adalah *mean reciprocal rank* (MRR). Makarand dan Parag (2018) menjelaskan cara menghitung MRR adalah dengan rumus sebagai berikut.

$$\boldsymbol{MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank\ i}} \quad \text{(II.15)}$$

Simbol $rank\ i$ mengacu pada posisi peringkat dokumen relevan pertama untuk *query* ke-i. Sebagai contoh, ketika ada empat *query* yang dimasukkan, hasil yang diinginkan muncul pada posisi 1, 5, 2, dan 10, maka nilai MRR-nya adalah (1/1 + 1/5 + 1/2 + 1/10)/4 = 0.45. Rentang nilai untuk MRR adalah antara 0 sampai 1 dengan nilai yang lebih besar berarti hasil yang lebih baik.

Bab III Analisis dan Rancangan Arsitektur Sistem

III.1 Tabel Posisi Penelitian

Tabel posisi penelitian berikut akan menjelaskan posisi penelitian ini jika dibandingkan dengan penelitian lain sejenis yang dilakukan pada domain yang sama yaitu pencarian musik dengan *query by humming*. Dari **Error! Reference source not found.** berikut dapat dilihat bahwa penelitian ini akan mengambil teknik pencocokan dari penelitian yang dilakukan oleh Makarand dan Parag (2018) yang juga membuat sistem pencarian musik dengan *query by humming*. Untuk dua tahapan lainnya, yaitu ekstraksi melodi dan filtrasi derau, teknik yang akan digunakan merupakan teknik yang belum pernah digunakan dalam pembuatan sistem pencarian musik dengan *query by humming*. Teknik yang akan digunakan untuk kedua tahapan tersebut adalah teknik ekstraksi melodi berbasis DNN-LSTM yang diperkenalkan oleh Cao dkk. (2020) dan teknik filtrasi derau dengan *Fourier series decomposition* dan *spectral subtraction* yang diperkenalkan oleh Siam dkk. (2019).

Tabel III.1 Tabel posisi penelitian

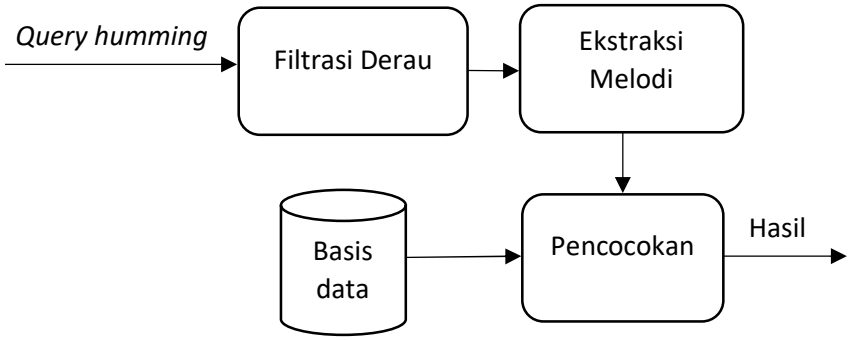
Peneliti	**Ekstraksi Melodi**	**Filtrasi Derau**	**Pencocokan**
Ghias dkk. (1995)	*Pitch tracking*	Tidak disebutkan	*Pattern matching*
Blackburn dan DeRoure (1998)	*Pitch tracking*	Tidak disebutkan	*Tree based search*
Shih dkk. (2003)	MFCC[4]	-	HMM[5] dengan GMM[6]
Tao dkk. (2009)	*Harmonic Product Spectral*	Tidak disebutkan	*Approximate String Matching*
Yang dkk. (2010)	*Pitch tracking*	Tidak disebutkan	*Note-based linear scaling* dan *recursive align*
Zhou dkk. (2017)	*Autocorrelation function*	*Smoothing* dan *pitch conversion*	DTW[7], HMM, LSH[8]
Makarand dan Parag (2018)	PRAAT[9]	-	*Unified algorithm*

[4] Mel-Frequency Cepstral Coefficients
[5] Hidden Markov Models
[6] Gaussian mixture models
[7] *Dynamic Time Warping*
[8] *Location Sensitive Hash*
[9] PRAAT. https://www.fon.hum.uva.nl/praat/

Koirala dkk. (2018)	AubioPitch[10]	*Auto tuning* dan *pitch normalization*	*Fast Dynamic Time Warping*
Penelitian ini	*Multi task learning* berbasis DNN-LSTM	*Fourier series decomposition* dan *spectral subtraction*	*Unified algorithm*

III.2 Arsitektur Sistem

Arsitektur sistem pencarian musik dengan *query by humming* yang akan dibangun dalam penelitian ini secara umum dapat dilihat pada Gambar III.1. Penjelasan lebih detail untuk setiap tahap dapat dilihat pada beberapa subbab berikut.



Gambar III.1 Rancangan arsitektur sistem

III.3 Filtrasi Derau

Seperti yang disampaikan oleh Makarand dan Parag (2018), tahap pembersihan data atau sinyal *query humming* sebelum masuk proses selanjutnya penting dilakukan untuk menambah akurasi hasil pencarian. Salah satu proses pembersihan data yang biasa dilakukan pada sinyal suara adalah dengan mereduksi derau. Rajini dkk. (2019) melakukan perbandingan pada beberapa teknik untuk melakukan reduksi derau yang dapat dikelompokkan menjadi dua metode, yaitu metode filtrasi dan *neural network*. Penelitian tersebut menghasilkan kesimpulan berupa metode filtrasi lebih baik dalam melakukan reduksi derau pada sinyal ucapan dibandingkan dengan metode *neural network* yang lebih kompleks dan membutuhkan waktu eksekusi yang lebih lama.

[10] Aubio. https://aubio.org

Teknik untuk mereduksi derau yang sudah dibuat pada umumnya digunakan untuk mereduksi derau pada sinyal yang berisi ucapan (*speech*). Siam dkk. (2019) memperkenalkan salah satu teknik filtrasi untuk mereduksi derau dengan *Fourier series decomposition* dan *spectral subtraction* yang sudah dicoba untuk domain identifikasi pembicara. Penelitian ini menghasilkan kesimpulan bahwa teknik yang diajukan memiliki hasil yang lebih baik dibandingkan beberapa teknik lain, yaitu *Wiener filter* dan *spectral subtraction*, dan terbukti dapat digunakan untuk menghasilkan sistem yang *robust* pada domain identifikasi pembicara. Cara kerja teknik tersebut secara umum sudah dijelaskan pada subbab II.3.2. Eksperimen untuk menerapkan teknik ini pada sistem pencarian musik dengan *query by humming* belum pernah dilakukan sebelumnya. Secara umum proses untuk melakukan filtrasi derau pada sinyal yang berisi ucapan dan sinyal yang berisi *query humming* sama saja karena baik sinyal ucapan maupun sinyal musik keduanya berasal dari vokal manusia dan antara sinyal utama dengan derau dapat dibedakan.

Pada penelitian yang dilakukan Siam dkk. (2019), jenis derau yang bisa ditangani dengan teknik *Fourier series decomposition* dan *spectral subtraction* adalah *white noise* atau derau yang memiliki intensitas yang sama pada frekuensi berbeda. Derau ini bunyinya seperti yang biasa kita dengar di perangkat radio pada frekuensi yang kosong atau tidak ada stasiun radio yang mengudara pada frekuensi tersebut. Bentuk gelombang sinyal yang memiliki derau ini dapat dilihat pada Gambar II.10 dan Gambar II.11. Derau jenis ini cenderung konstan mengisi dari awal sampai akhir sebuah file audio dan biasa dihasilkan dari proses *grounding* yang tidak tepat pada perangkat perekaman audio.

III.4 Ekstraksi Melodi

Setelah sinyal suara yang menjadi *query* melalui proses filtrasi derau, langkah selanjutnya adalah melakukan ekstraksi melodi untuk mendapatkan kelas nada atau not dari setiap fragmen suara. Teknik ekstraksi melodi yang akan digunakan pada penelitian ini adalah *multi task learning* berbasis DNN-LSTM yang diperkenalkan oleh Cao dkk. (2020). Ada dua tugas yang dilakukan dalam metode ini, yaitu

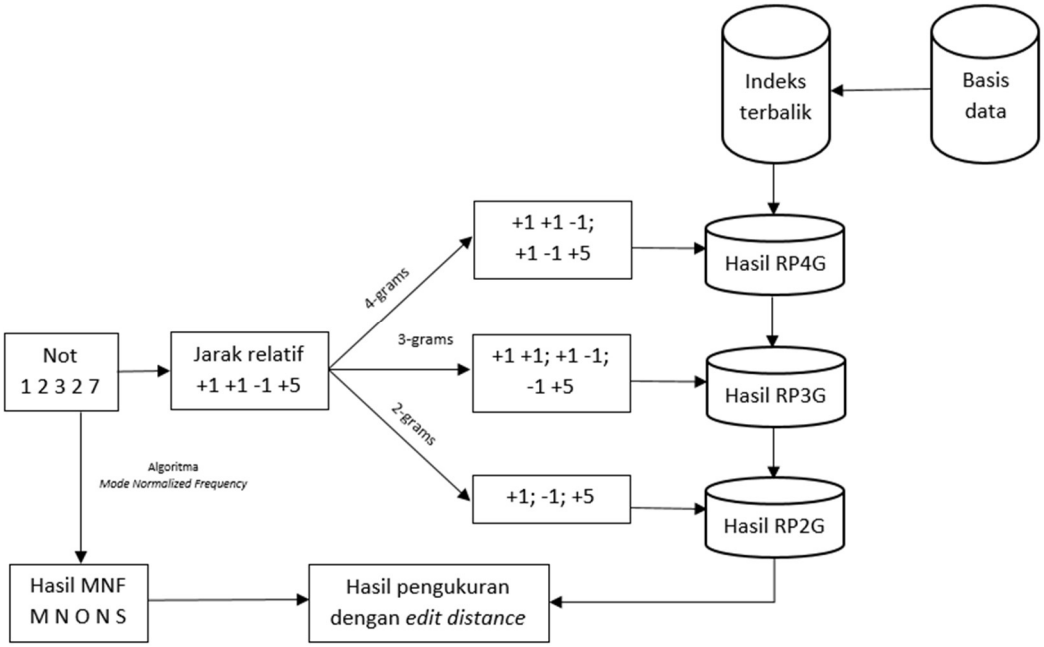
estimasi nada dan deteksi suara. Teknik ekstraksi melodi ini terbukti dapat menghasilkan akurasi yang lebih tinggi dan kemampuan generalisasi yang lebih baik. Cara kerja teknik tersebut secara umum sudah dijelaskan pada subbab II.4.1.

III.5 Pencocokan

Setelah didapatkan kelas nada dari *query* yang menjadi masukan, langkah selanjutnya adalah melakukan pencocokan antara *query* dengan data musik yang disimpan di basis data untuk mencari musik yang paling mirip dengan *query*. Sebelum masuk proses ini, semua musik yang disimpan di basis data perlu terlebih dahulu dilakukan proses yang sama dengan *query*, yaitu dilakukan filtrasi derau untuk mereduksi derau yang mungkin ada pada data musik dan ekstraksi melodi untuk mendapatkan kelas nada dari setiap musik dalam basis data.

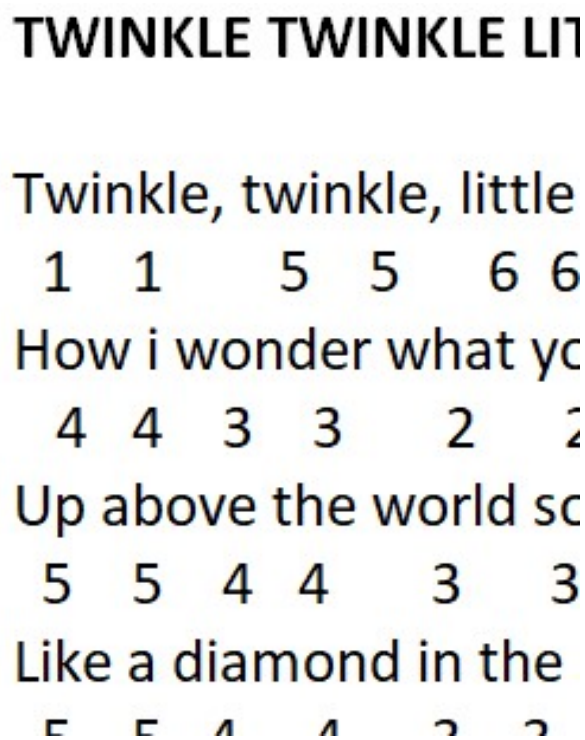
Teknik pencocokan yang akan digunakan pada penelitian ini adalah *unified algorithm* yang diperkenalkan oleh Makarand dan Parag (2018). Sistem yang dibuat pada penelitian tersebut juga yang akan menjadi *base-line system query by humming* yang akan menjadi pembanding dengan sistem yang akan dibuat pada penelitian ini. Cara kerja teknik tersebut secara umum sudah dijelaskan pada subbab II.2.7. Secara visual, teknik tersebut dapat diilustrasikan seperti pada Gambar III.2.

Teknik ini dapat menyelesaikan permasalahan berupa perbedaan nada dasar yang mungkin terjadi antar pengguna, di mana lagu yang sama bisa digumamkan ke dalam dua *query* yang berbeda karena tiap pengguna memakai nada dasar yang berbeda. Hal ini dapat ditangani oleh teknik ini karena yang dilihat adalah jarak relatif antar nada dan nada yang paling sering muncul sehingga teknik ini dapat berjalan baik dengan asumsi naik turunnya nada yang digunakan benar sesuai dengan seharusnya. Selain itu, teknik ini juga dapat menyelesaikan permasalahan berupa perbedaan tempo antar pengguna karena tempo yang merupakan selang waktu antar nada tidak diperhatikan dalam proses pencocokan. Nada yang berulang secara berurutan juga dihilangkan dan hanya diambil satu nada saja untuk menyederhanakan pencarian.



Gambar III.2 Alur kerja teknik pencocokan dengan *unified algorithm*

Masalah lain yang ada dalam pembangunan sistem pencarian musik dengan *query by humming* adalah ketika pengguna membuat gumaman yang tidak kontinyu. Tidak kontinyu di sini berarti gumaman lagu yang dijadikan *query* memiliki urutan nada yang tidak sesuai dengan seharusnya. Hal ini biasa terjadi karena pengguna tidak bisa mengingat urutan bagian lagu dengan baik dan hanya mengingat sebagian lagu yang nadanya membekas di ingatan. Sebagai contoh, Gambar III.3 adalah rangkaian not dari lagu Twinkle Twinkle Little Star. Ketika pengguna memasukkan *query* gumaman yang tidak kontinyu, semisal not yang diekstrak dari *query* adalah 1 1 5 5 6 6 5 5 5 4 4 3 3 2 yang merupakan not dari baris ke-1 yang dilanjutkan dengan not dari baris ke-3, maka hasil pencarian tidak akan terlalu baik karena sistem tidak melihat kesamaan yang utuh antara gumaman dengan musik dalam basis data.



Gambar III.3 Rangkaian not lagu Twinkle Twinkle Little Star[11]

Salah satu ide untuk menyelesaikan masalah ini adalah dengan menguraikan *query* menjadi beberapa bagian dengan panjang yang bervariasi. Seperti kita sadari, ketika kita tidak bisa mengingat urutan bagian lagu dengan sempurna, kita tetap akan mengingat lagu tersebut per bagian secara utuh, seperti bagian intro saja atau reff saja. Dengan kita menguraikan *query* menjadi beberapa bagian dan melakukan pencarian ke basis data dengan bagian *query* tersebut, maka nilai kesamaannya bisa lebih tinggi. Seperti contoh *query* pada paragraf sebelumnya, jika kita menguraikan *query* tersebut menjadi dua bagian dengan panjang masing-masing adalah 7 not, maka kita mendapatkan dua buah *query* yang masing-masing merupakan not dari baris ke-1 dan ke-3 dari lagu sebenarnya. Setiap bagian *query* digunakan untuk melakukan pencarian dari basis data dan hasil dari setiap pencarian disatukan dengan cara diambil hasil yang sama yang memiliki nilai kesamaan tinggi.

Untuk mengukur kualitas dari sistem pencarian musik *query by humming* yang dibuat, akan digunakan dua macam cara pengukuran, yaitu *top 3/5/10 hit ratios* dan

[11] https://notangka-musik.blogspot.com/2018/04/not-angka-pianika-lagu-twinkle-twinkle.html

mean reciprocal rank (MRR). Dari hasil keluaran algoritma pencocokan, akan dilihat berapa persen *query* yang diprediksi dengan benar dan masuk dalam urutan 3, 5, atau 10 teratas untuk mengetahui seberapa baik sistem yang sudah dibuat untuk *query* yang berasal dari satu sumber tertentu, semisal dari satu orang. Selain itu, dihitung juga nilai MRR yang menunjukkan akurasi dari suatu algoritma untuk keseluruhan *query* yang digunakan. Cara penghitungan MRR dijelaskan pada subbab II.5.

III.6 Dataset

Dataset yang akan digunakan diambil dari forum Music Information Retrieval Evaluation eXchange (MIREX)[12] untuk tugas *Query-by-Singing/Humming* (QBSH). Ada dua dataset yang disediakan dalam forum ini, dataset pertama adalah korpus MIR-QBSH dari Roger Jang yang terdiri dari 4.431 *query* dengan 48 musik dalam basis data sedangkan dataset kedua adalah korpus IOACAS yang terdiri dari 759 *query* dengan 298 musik dalam basis data.

[12] MIREX https://www.music-ir.org/mirex/wiki/MIREX_HOME

Bab IV Eksperimen dan Analisis Hasil

IV.1 Eksperimen

TBD

IV.2 Hasil Eksperimen

TBD

IV.3 Analisis Hasil Eksperimen

TBD

Bab V Kesimpulan dan Saran

V.1 Kesimpulan

TBD

V.2 Saran

TBD

DAFTAR PUSTAKA

Blackburn, S., dan DeRoure, D. (1998): A tool for content based navigationof music, *Proceedings of the 6th ACM International Conference on Multimedia, MULTIMEDIA 1998*. https://doi.org/10.1145/290747.290802

Cao, Z., Feng, X., dan Li, W. (2020): A multi-task learning approach for melody extraction, *Lecture Notes in Electrical Engineering*, **635**, 53–65. https://doi.org/10.1007/978-981-15-2756-2_5

Ghias, A., Logan, J., Chamberlin, D., dan Smith, B. C. (1995): Query by humming, 231–236. https://doi.org/10.1145/217279.215273

Khan, N. A., dan Mushtaq, M. (2011): Open issues on query by humming, *4th International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2011*, 147–152. https://doi.org/10.1109/ICADIWT.2011.6041417

Koirala, P., Chapagain, M., Pantha, N., dan Adhikar, N. B. (2018): Effects of Auto Tuning and Pitch Normalization on Query by Humming, **01**(02), 1–6.

Kurzekar, P. K., Deshmukh, R. R., Waghmare, V. B., dan Shrishrimal, P. P. (2014): A Comparative Study of Feature Extraction Techniques for Speech Recognition System, *International Journal of Innovative Research in Science, Engineering and Technology*. https://doi.org/10.15680/ijirset.2014.0312034

Makarand, V., dan Parag, K. (2018): Unified algorithm for melodic music similarity and retrieval in query by humming, *Advances in Intelligent Systems and Computing*, **673**, 373–381. https://doi.org/10.1007/978-981-10-7245-1_37

Makarand, V., dan Sahasrabuddhe, H. V. (2014): Novel approach for music search using music contents and human perception, *Proceedings - International Conference on Electronic Systems, Signal Processing, and Computing Technologies, ICESC 2014*, 1–6. https://doi.org/10.1109/ICESC.2014.9

Rajini, G. K., Harikrishnan, V., Jasmin Pemeena Priyadarisini, M., dan Balaji, S. (2019): A research on different filtering techniques and neural networks methods for denoising speech signals, *International Journal of Innovative Technology and Exploring Engineering*, **8**(9 Special issue 2), 503–511. https://doi.org/10.35940/ijitee.I1107.0789S219

Shih, H. H., Narayanan, S. S., dan Kuo, C. C. J. (2003): Multidimensional humming transcription using a statistical approach for query by humming systems, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. https://doi.org/10.1109/icme.2003.1221329

Siam, A. I., El-khobby, H. A., Elnaby, M. M. A., Abdelkader, H. S., dan El-samie, F. E. A. (2019): A Novel Speech Enhancement Method Using Fourier Series Decomposition and Spectral Subtraction for Robust Speaker Identification, *Wireless Personal Communications*, (0123456789). https://doi.org/10.1007/s11277-019-06453-4

Tao, L., Xianglin, H., Lifang, Y., dan Pengju, Z. (2009): Query by humming: Comparing voices to voices, *Proceedings - International Conference on Management and Service Science, MASS 2009*, 5–8.

https://doi.org/10.1109/ICMSS.2009.5305356
Tseng, Y. H. (1999): Content-based retrieval for music collections, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999*. https://doi.org/10.1145/312624.312675
Wang, C. C., dan Jang, J. S. R. (2015): Improving query-by-singing/humming by combining melody and lyric information, *IEEE/ACM Transactions on Audio Speech and Language Processing*. https://doi.org/10.1109/TASLP.2015.2409735
Yang, J., Liu, J., dan Zhang, W. Q. (2010): A fast query by humming system based on notes, *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, 2898–2901.
Zhou, S., Zhao, Z., Shi, P., dan Han, M. (2017): Research on matching method in humming retrieval, *Proceedings of 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference, ITOEC 2017*, **2017-Janua**, 516–520. https://doi.org/10.1109/ITOEC.2017.8122349

LAMPIRAN