



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Literature Review

Penelitian komposer musik lainnya pernah dibuat oleh Ronald Wilson [3], yang membuat komposer alat – alat musik fisik populer, seperti piano, *bass*, dan *string* menggunakan MATLAB. Komposer tersebut dapat menyusun musik seperti yang dikehendaki *user*, dengan fitur pendukung lainnya seperti tempo dan volume, serta dilengkapi *Graphic User Interface (GUI)* yang *user friendly*. Suara – suara yang sudah diinput kemudian digabungkan sehingga menghasilkan musik dari berbagai suara. Hasil musik juga diplot dalam bentuk grafik dan menghasilkan *file output* WAV. Dalam penelitian ini, pilihan musik terbatas dan tidak bisa menambahkan alat musik lain.

Selain itu Tim Thompson juga pernah membuat komposer musik [4]. Pada komposer ini *database* musik telah ada sebelumnya, sehingga *user* dapat memasukkannya langsung ke dalam *sequence*. *Database* musik yang ada berupa *file* drum yang dapat dipilih dan dimainkan. Fitur pendukungnya yaitu *on / off sequence*, volume, dan tempo. Kekurangan dalam penelitian ini adalah hasil *output* berbentuk *file* MIDI, sudah tidak umum digunakan dan tidak dapat diolah untuk proses *denoising*.

Penelitian komposer musik lainnya juga pernah dibuat oleh Kai Yang dan Xi Zhou [5]. Mereka merancang sebuah program musik untuk membuat musik MIDI. Program ini terdapat fitur – fitur untuk pelengkapannya, seperti volume, *rhythm*, *play*, *stop*, dan lainnya. Musik ditentukan oleh nada-nada yang

diinput dalam bentuk angka. Misalnya angka 5 berarti sol, angka 2 berarti re, begitu juga dengan nada lainnya.

Penelitian tentang *denoising* yang serupa telah dilakukan oleh Neeraj Kumari dan Shelly Chugh. Proses reduksi *noise* melibatkan beberapa jenis *wavelet* dengan pendekatan *soft* dan *hard thresholding* serta kombinasi level dekomposisi. Pengamatan analisa dilakukan dengan sinyal domain waktu dan spektogram [6].

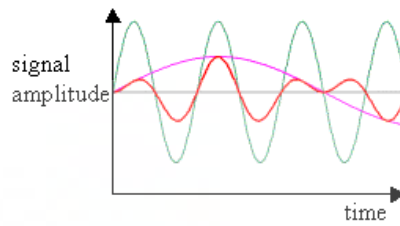
Pada penelitian ini, musik dapat diatur sesuai dengan kehendak *user* dengan mengatur *sequence* pada GUI yang ada. Selain itu, perekaman suara tidak perlu menghasilkan nada, tetapi juga bisa suara seperti *beatbox*, sehingga tidak memerlukan pemahaman lebih mengenai musik. Kemudian kelebihan yang lainnya, dari hasil musik tersebut, audio dapat direduksi *noise* dengan menggunakan transformasi *wavelet*, sehingga hasil suara dapat lebih jernih.

## 2.2. Pengolahan Suara dan Audio

### 2.2.1. Sinyal Suara

#### A. Sinyal Suara dalam Domain Waktu

Sinyal suara merupakan besaran fisik yang berubah terhadap waktu. Besaran fisik merupakan intensitas bunyi sedangkan variabel bebasnya adalah waktu. Ketika diplot terdapat salah satu sumbu dengan variabel waktu dan sumbu lainnya yaitu amplitudo. Amplitudo pada sinyal domain ini menunjukkan keras atau lemahnya sinyal [7].

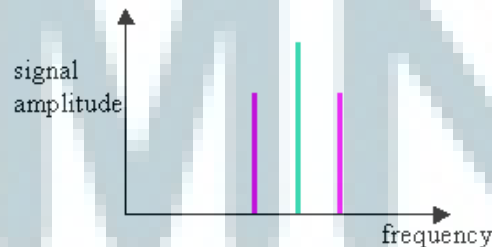


**Time domain**

Gambar 2.1. Sinyal Suara dalam Domain Waktu [8]

### **B. Sinyal Suara dalam Domain Frekuensi**

Sinyal suara dalam domain frekuensi merupakan gabungan dari satu atau lebih sinyal sinusoidal dengan frekuensi dan intensitas yang berbeda-beda. Dengan kata lain, sinyal suara ini dipresentasikan sebagai intensitas dari komponen frekuensi penyusunnya. Contohnya not dalam musik, suatu bunyi dihasilkan dengan memainkan satu atau beberapa nada bersamaan. Spektrum pada domain ini menunjukkan frekuensi apa yang muncul. Frekuensi ini menunjukkan tingkat perubahan. Sedangkan sumbu yang lain menunjukkan magnitudo. Magnitudo ini menunjukkan tinggi rendahnya sinyal yang diterima. Keras atau lemahnya sinyal tidak mempengaruhi frekuensi yang di dalamnya [7].



**Frequency domain**

Gambar 2.2. Sinyal Suara dalam Domain Frekuensi [8]

Sinyal suara dalam domain waktu dapat diubah ke dalam domain frekuensi dengan menggunakan transformasi *fourier*. Selain itu, sinyal suara dalam domain frekuensi dapat diubah kembali ke dalam domain waktu dengan menggunakan transformasi balik *fourier* atau *inverse fourier transform (IFT)*.

### 2.2.2. Audio Digital

Suara yang berada pada jangkauan pendengaran manusia disebut audio, dan suara di luar jangkauan pendengaran manusia dapat dikatakan sebagai *noise* [9].

Audio digital merupakan versi digital dari suara analog. Proses berubahan dari analog ke digital memerlukan konverter yang dinamakan *Analog to Digital Converter (ADC)*. Audio digital merupakan sampel suara. Kualitas perekamannya bergantung pada seberapa sering sampel diambil. Contoh *sampling* yang sering digunakan dalam multimedia adalah 44.1 kHz, 22.05 kHz, dan 11.025 kHz dengan ukuran sampel 8 bit dan 16 bit. Ukuran sampel 8 bit menyediakan 256 unit deskripsi jarak dinamis atau amplitudo. Ukuran *file* audio digital bergantung pada angka *sampling*, resolusi dan *channel* (*stereo* atau *mono*) [10].

### 2.2.3. Channels

*Channels* merupakan banyaknya suara yang disimpan dalam sebuah *file* audio digital. Biasanya ada dua tipe *channels* pada format WAV, yaitu

*mono* dan *stereo*. *Channels* *mono* mempunyai satu buah suara yang tersimpan pada suara kiri atau suara kanan, atau gabungan kedua buah suara tersebut. Sedangkan *stereo* mempunyai dua buah suara yang tersimpan yaitu suara kiri dan suara kanan.[11]

#### 2.2.4. Frekuensi *Sampling*

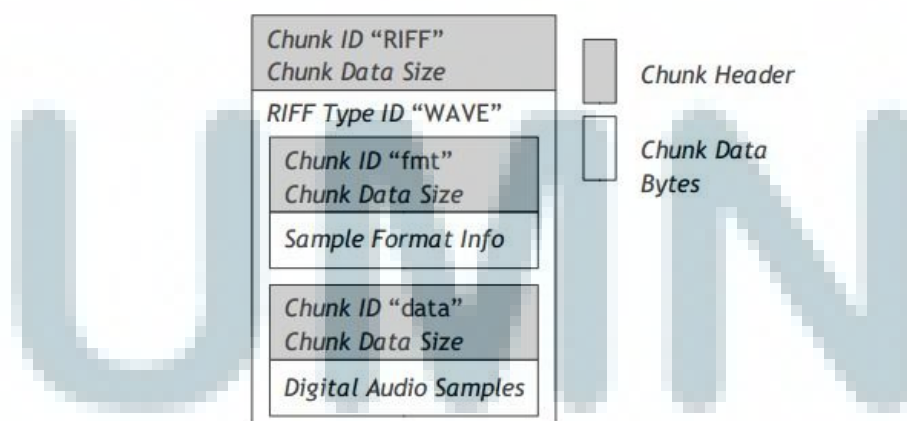
Sebuah teknik memungkinkan sinyal audio diubah dan diproses sehingga menjadi lebih baik. Teknik ini memungkinkan perubahan sinyal analog menjadi bit-bit digital. Teknik ini disebut teknik *sampling*, sehingga jauh lebih baik, *noise* dapat direduksi dan dapat diproses dengan mudah.

Frekuensi *sampling* (*sample rate*) merupakan jumlah *sample* per detik dalam suatu suara. Contoh, jika frekuensi *sampling* adalah 44.100 Hertz, dan ada rekaman berdurasi 30 detik akan berisi 1.323.000 *samples* (untuk kualitas CD) dan 22.050 Hertz (untuk pembicaraan biasa). Menurut teorema Nyquist, frekuensi *sampling* minimum adalah dua kali *bandwith* dari sinyal yang di *sampling* untuk mencegah terjadinya *aliasing* (pergeseran frekuensi tinggi menjadi lebih rendah diakibatkan interval *sampling* yang terlalu besar). Ini adalah batas minimum dari frekuensi sampel agar nantinya cuplikan yang diambil menunjukkan bentuk sinyal yang asli (analog). Cuplikan lebih besar biasanya lebih baik karena cuplikan akan lebih menggambarkan sinyal yang asli [12].

### 2.2.5. File WAV

Format berkas WAV (WAVE) merupakan berkas standar format untuk penyimpanan data audio digital dalam *Windows*, biasanya menggunakan coding *Pulse Code Modulation* (PCM). *File* WAV mirip dengan *file* AIFF yaitu *file* audio yang digunakan di komputer Mac. Hampir setiap program pemutar atau pengolah audio menyediakan dukungan untuk membuka dan/atau menyimpan ke dalam format ini. WAV tidak terkompres sehingga seluruh sampel audio disimpan semuanya [13].

Berkas WAVE dapat mengandung *string* karakter. *String* ini disimpan dalam format dimana *byte* pertama menyatakan jumlah karakter dari *string* dan kemudian diikuti oleh isi *string* dalam representasi ASCII. Struktur WAVE menggunakan struktur standar RIFF (*Resource Interchange File Format*). Format ini mengelompokkan bagian-bagian dari *file* menjadi kelompok-kelompok terpisah yang kemudian disimpan dalam *chunk* yang terpisah.



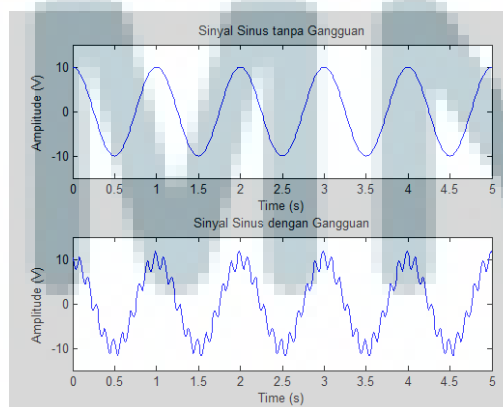
Gambar 2.3. Dasar Struktur Format WAV [14]

Gambar 2.3. menampilkan struktur format WAV paling sederhana. Struktur ini adalah *chunk* RIFF yang mengandung sub *chunk* “fmt” yang menyimpan data atau info tentang format sampel audio dan sebuah sub *chunk* “data” yang berisi sampel audio sebenarnya. Kualitas produksi tipe audio ini bergantung pada *sampling rate* (banyaknya sampel per-detik).

### 2.2.6. Noise

*Noise* merupakan suatu sinyal pengganggu yang menyebabkan suatu sinyal mengalami gangguan. Gangguan yang diakibatkan *noise* dapat merusak bahkan mengubah sinyal informasi. Hal-hal yang mempengaruhi *noise* bisa dari faktor *hardware*, *software*, dan lingkungan [15]

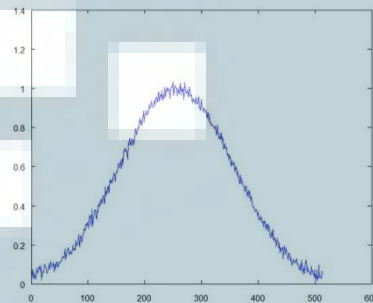
*Noise* dapat merubah bentuk gelombang asli. *Noise* juga dapat merusak bentuk sinyal asli, menambah atau mengurangi amplitudo, mempercepat atau memperlambat waktu, dan bentuk perubahan lainnya. Gambar 2.4. merupakan perbedaan sinyal sinusoidal asli tanpa gangguan dengan sinyal sinusoidal dengan *noise* atau gangguan.



Gambar 2.4. Perbandingan sinyal tanpa gangguan dengan adanya gangguan



Pada Gambar 2.4. terlihat *noise* dapat mengubah bentuk gelombang asli, sehingga tidak nampak informasi sebenarnya dari gelombang sinus tersebut. Pada penelitian ini digunakan *white gaussian noise* yang merupakan suatu *noise* dengan kerapatan spektral daya yang merata pada seluruh komponen frekuensinya. Gambar 2.5. menunjukkan contoh spektrum *white gaussian noise*, yang artinya dalam penambahan *noise* berdasarkan distribusi normal.[15]



Gambar 2.5. Contoh bentuk *white gaussian noise*

Pada reduksi *noise*, ditambahkan *noise* terlebih dahulu dengan satuan dB. Semakin besar nilai yang ditambahkan semakin kecil *noise* yang ditambahkan, sedangkan semakin kecil nilai yang ditambahkan semakin besar *noise* yang ditambahkan.

### 2.2.7. Equalizer

*Equalizer* adalah perangkat umum yang digunakan dibanyak jenis pengolahan sinyal. Melalui *equalizer user* dapat mengubah karakter suara secara total atau sebagian. Kegunaannya adalah dapat mengubah karakteristik suara dan biasanya digunakan untuk mengontol frekuensi respon dari *sound*

*system. Equalizer* mampu melakukan perbaikan pada kualitas suara yang disebabkan oleh adanya keterbatasan dari peralatan audio. Metode ini mengurangi distorsi suara karena tidak bagusnya suara saat perekaman sehingga dapat menghasilkan output audio yang bagus. Konsep *equalizer* juga diimplementasi pada kehidupan sehari-hari seperti mengurangi reduksi *noise* pada telepon, hubungan frekuensi radio dan komunikasi digital. [16]

Meningkatkan kejelasan suara instrumen. Tiap instrumen musik memiliki suara yang khas dan berkarakter. Suara seperti *hi-hat*, *kick*, dapat terlihat cerah atau mantap jika diatur frekuensinya. Dengan kata lain, *equalizer* dapat membuat karakter suara menjadi lebih jelas.

Dalam *equalizer* biasanya terdapat fungsi-fungsi sebagai berikut.

- a. *Low Pass / High Pass*, digunakan untuk menghilangkan frekuensi-frekuensi di bawah atau di atas frekuensi yang dipilih.
- b. Plot, digunakan untuk menampilkan hasil sinyal setelah filter diubah[17]

Setiap band frekuensi dalam *equalizer* grafik merupakan jenis *Band Pass Filter*. *High Pass Filter* untuk memotong frekuensi tinggi, sedangkan *Low Pass Filter* untuk memotong frekuensi rendah. Dasar pembuatan sebuah *equalizer* adalah frekuensi yang dibuat berkisar antara 20 Hz – 20 kHz, dimana rentang frekuensi ini adalah frekuensi yang dapat diterima oleh pendengaran manusia. Frekuensi yang berada di bawah atau di atas rentang frekuensi tersebut tidak dapat didengarkan manusia. Pada *equalizer* grafik ini berfungsi untuk mengkompensasi respon frekuensi dari speaker akibat

peletakan speaker, memperbaiki masalah akibat ruangan, dan mereduksi *feedback*. *Equalizer* memiliki jangkauan frekuensi yang detail dan lengkap karena terbagi atas beberapa oktav, mulai dari 1 oktaf hingga 1/3 oktaf.

### 2.3. Transformasi *Wavelet*

Sinyal analog dan digital pada umumnya direpresentasikan dalam domain waktu. Dalam domain waktu ini, sulit dilakukan analisis frekuensi suatu sinyal. Transformasi digunakan untuk mengubah suatu sinyal dari domain waktu ke domain frekuensi, sehingga mempermudah analisis maupun perhitungan.

Sekitar tahun 1980 pertama kali ditemukannya transformasi *wavelet*, dimana transformasi *wavelet* ini digunakan sebagai alternatif pengganti Short Time Fourier Transform untuk analisa sinyal. Transformasi *wavelet* dapat memberikan informasi tentang kombinasi skala dan frekuensi. *Wavelet* adalah “gelombang singkat” dengan energi terpusat pada saat tertentu yang merupakan kelas dari suatu fungsi yang digunakan untuk melokalisasi suatu fungsi dalam ruang dan skala. *Wavelet* telah dipakai dalam analisa sinyal kawasan waktu-frekuensi dalam pemrosesan sinyal, aproksimasi fungsi, aproksimasi dalam penyelesaian persamaan diferensial parsial dan sebagainya. Salah satu kegunaan dari *wavelet* yaitu dapat digunakan untuk reduksi *noise* pada sinyal suara. [17] Analisis *wavelet* berasal dari fungsi yang dinamakan *mother wavelet*, yang dituliskan sebagai berikut.

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \Psi\left(\frac{t-\tau}{s}\right) \quad (2.4)$$

S merupakan proses *scaling* atau pelebaran dari fungsi dasar,  $\tau$  merupakan proses translasi atau *shifting* terhadap posisi dari sumbu x.  $\Psi\left(\frac{t-\tau}{s}\right)$  merupakan *mother wavelet*,  $\frac{1}{\sqrt{s}}$  merupakan normalisasi untuk memastikan energi dalam berbagai *scaling* tetap sama. Translasi adalah lokasi jendela modulasi saat digeser sepanjang sinyal, berhubungan dengan informasi waktu. *Scaling* atau skala berhubungan dengan frekuensi. Skala tinggi (frekuensi rendah) berhubungan dengan informasi global dari sebuah sinyal, sedangkan skala rendah (frekuensi tinggi) berhubungan dengan informasi detail [18].

Dalam praktek, kebanyakan sinyal berada dalam domain waktu, sehingga sinyal selalu dinyatakan dalam fungsi waktu, sehingga hasilnya selalu dalam koordinat waktu dan amplitudo. Salah satu perbedaan penting yaitu transformasi *wavelet* membolehkan penempatan waktu dalam komponen-komponen frekuensi yang berbeda dari sinyal yang diberikan. Transformasi Fourier hanya dapat menangkap informasi apakah suatu sinyal memiliki frekuensi tertentu ataukah tidak, tapi tidak dapat menangkap dimana frekuensi itu terjadi. Jika Transformasi Fourier hanya memberikan informasi tentang frekuensi suatu sinyal, maka transformasi *wavelet* memberikan informasi tentang kombinasi skala dan frekuensi. Sebagai ilustrasi pada konser musik, Transformasi Fourier hanya bisa mengatakan apakah suatu 'nada' tertentu muncul, tapi tidak dapat mengatakan berapa kali dan kapan nada itu muncul. Selain itu, Transformasi Fourier berdasarkan pada basis sin dan cos yang bersifat periodik dan kontinu, sehingga sulit jika ingin melakukan perubahan hanya pada posisi tertentu dan pasti akan mempengaruhi posisi-posisi lainnya.

Dengan kata lain, transformasi *Wavelet* merupakan salah satu sarana yang dapat digunakan untuk menganalisis (meneliti) sinyal-sinyal non-stasioner. Analisis *Wavelet* dapat digunakan untuk menunjukkan kelakuan sementara (temporal) pada suatu sinyal, misalnya dalam bidang geofisika (sinyal seismik), fluida, medik dan lain sebagainya. Metode Transformasi *Wavelet* ini dapat digunakan untuk menapis data atau meningkatkan mutu kualitas data; dapat juga digunakan untuk mendeteksi kejadian-kejadian tertentu serta dapat digunakan untuk pemampatan data [18].

Reduksi *noise* pada sinyal suara dengan transformasi *wavelet* yaitu melakukan transformasi pada sinyal suara sehingga didapat koefisien *wavelet*, kemudian dilakukan *threshold* (melewatkan pada ambang tertentu) koefisien *wavelet* sehingga *noise* dapat dikurangi. Kemudian melakukan transformasi *wavelet inverse* sehingga didapat sinyal suara.

Karakteristik umum sistem *wavelet* adalah sebagai berikut:

1. Waktu kompleksitasnya bersifat linear. Transformasi *wavelet* dapat dilakukan dengan sempurna dengan waktu yang bersifat linear.
2. Ekspansi *wavelet* memberikan lokalisasi sinyal pada waktu dan frekuensi.
3. Perhitungan koefisien sinyal dapat dilakukan dengan efisien. Secara praktis, koefisien-koefisien *wavelet* kebanyakan bernilai kecil atau nol. Kondisi ini sangat memberikan keuntungan terutama dalam bidang kompresi.

Proses transformasi *wavelet* dapat dilakukan dengan konvolusi atau dengan proses perataan dan pengurangan secara berulang. Pada proses transformasi *wavelet* ini dilakukan proses dekomposisi dan aproksimasi. Proses ini berdasarkan pada fungsi translasi  $\psi_{j,k}(t)$  dan fungsi skala  $\phi_{j,k}(t)$  sehingga dihasilkan

$$f(t) = \sum_k c_{jN}[k] \phi_{jN,k}(t) + \sum_k \sum_{j=1}^{jN} d_j[k] \psi_{j,k}(t) \quad (2.5)$$

dimana N adalah jumlah level dekomposisi. Koefisien fungsi skala  $c_j[k]$  disebut dengan aproksimasi yang menunjukkan frekuensi rendah dari sinyal (*Low Pass Filter*). Sedangkan koefisien fungsi *wavelet* lainnya  $d_j[k]$  disebut dengan detail yang menunjukkan bagian frekuensi tinggi dari sinyal (*High Pass Filter*). Koefisien-koefisien dari kombinasi linier disebut Transformasi *Wavelet* Diskrit (TWD) atau *Discrete Wavelet Transform* (DWT) [18].

### 2.3.1. *Discrete Wavelet Transform* (DWT)

*Discrete Wavelet Transform* (DWT) bekerja mentransformasikan sinyal yang berbentuk diskrit. Pengaplikasian DWT dalam penelitian ini dengan menggunakan data dan informasi input berupa sinyal suara yang nantinya akan diubah ke dalam bentuk diskrit.

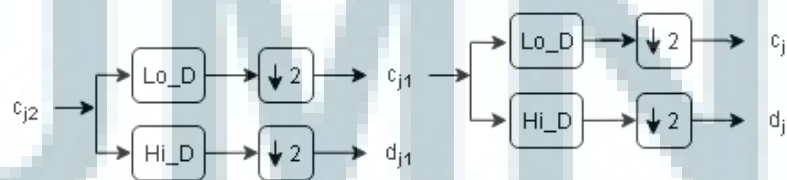
*Discrete wavelet transform* dianggap relatif lebih mudah dalam hal pengimplementasiannya. Prinsipnya adalah bagaimana cara mendapatkan representasi waktu dan skala dari sebuah sinyal menggunakan teknik pemfilteran digital dan operasi *up-sampling* atau *down-sampling*. Dalam

penelitian ini, DWT lebih mudah dalam penggunaannya dibandingkan *continuous wavelet transform* karena representasi sinyal yang dihasilkan sering berlebihan.

Sinyal pertama-tama dilewatkan pada rangkaian *high-pass* dan *low-pass filter*, kemudian setengah dari masing-masing keluaran diambil sebagai sampel melalui operasi *down-sampling*. Proses ini disebut sebagai proses dekomposisi satu tingkat. Keluaran dari *low-pass filter* digunakan sebagai masukan diproses dekomposisi yang diinginkan. Gabungan dari keluaran-keluaran *high-pass filter* dan satu keluaran *lowpass filter* yang terakhir disebut sebagai koefisien *wavelet*, yang berisi informasi sinyal hasil transformasi yang telah terkompresi [18].

### 2.3.2. Proses Dekomposisi

Proses ini merupakan proses memecah suatu sinyal menjadi beberapa komponen dengan resolusi lebih rendah. Operasi dasar pada proses dekomposisi adalah *filtering* dan *down sampling*. Contoh diagram dekomposisi *wavelet* dapat dilihat pada Gambar 2.6 [18].

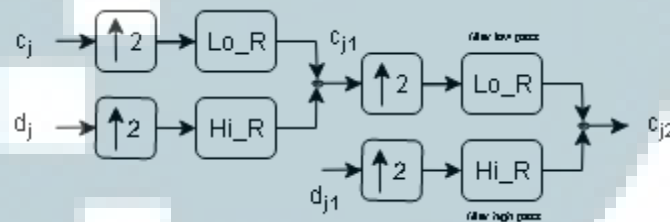


Gambar 2.6. Proses Dekomposisi

Operasi *down-sampling* berguna menghilangkan informasi sinyal yang berlebihan, sehingga transformasi *wavelet* menjadi salah satu metode kompresi data yang paling handal.

### 2.3.3. Proses Rekonstruksi

Proses ini mempunyai prinsip kerja kebalikan dari prinsip kerja dekomposisi. Cara kerja ini terlihat pada proses *filtering* dan *up-sampling*, yang merupakan dua operasi dasar pada proses rekonstruksi. Diagram proses rekonstruksi dapat dilihat pada Gambar 2.7.



Gambar 2.7. Proses Rekonstruksi

Sehingga secara matematis proses dekomposisi *wavelet* dapat dinyatakan oleh persamaan sebagai berikut:

### 2.3.4. Thresholding

Proses ini merupakan proses dimana reduksi *noise* terjadi. Koefisien-koefisien detail audio yang diperoleh melalui proses dekomposisi akan dilewatkan dan melalui proses *filtering* yang disebut dengan fungsi *thresholding*. Proses reduksi *noise* dilakukan setelah mendapatkan hasil



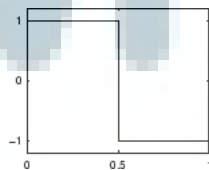
transformasi yang berupa nilai koefisien aproksimasi dan tiga orientasi koefisien detail dari tiap level dekomposisi.

Pada proses *thresholding* koefisien detail akan dibandingkan dengan suatu nilai *threshold t*, dan akan mendapat perlakuan sesuai dengan fungsi *threshold* yang digunakan. Hasil akhir reduksi *noise*, selain dipengaruhi oleh pemilihan filter, juga dipengaruhi oleh *threshold*, sehingga perlu dipilih metode pencarian nilai *threshold* yang tepat.

### 2.3.5. Wavelet Haar

Transformasi *wavelet* Haar merupakan *wavelet* tertua dan sederhana yang ditemukan pada tahun 1909. Jenis *wavelet* ini dikenal sebagai *mother wavelet* atau *wavelet* induk yang digunakan sejak pertama kali secara intensif. *Wavelet* ini termasuk dalam kategori ortogonal. Oleh karena itu peneliti memilih jenis *wavelet* ini karena merupakan jenis *wavelet* yang paling awal. Jenis dari transformasi Haar adalah sebagai berikut

1. Tidak membutuhkan perkalian. Hanya membutuhkan penambahan dan banyak elemen bernilai nol di matriks Haar, jadi waktu komputasi menjadi singkat.
2. Dapat digunakan untuk menganalisa sinyal., termasuk dalam properti orthogonal, maka sinyal input frekuensi dapat dianalisa.



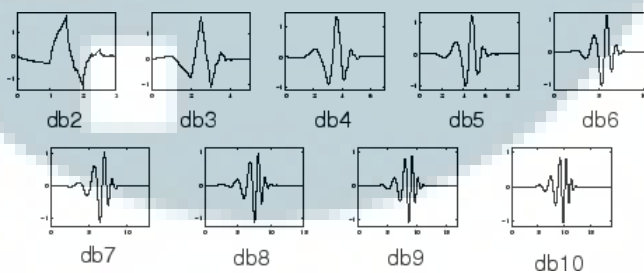
Gambar 2.8 *Wavelet* Haar [19]

Gambar 2.8 merupakan bentuk *wavelet* Haar. Fungsi *mother wavelet* Haar dapat dideskripsikan sebagai berikut.

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2}, \\ -1 & \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (2.8)$$

### 2.3.6. *Wavelet* Daubechies

Filter *wavelet* yang digunakan dalam tugas akhir ini adalah filter *wavelet* Daubechies. Transformasi ini merupakan metode yang digunakan untuk mengolah sinyal nonstationer dan untuk menganalisa sinyal yang selalu mengalami perubahan. Transformasi *wavelet* Daubechies dapat memberikan representasi waktu dan frekuensi secara bersamaan sehingga dapat mempresentasikan sinyal yang dimaksud.



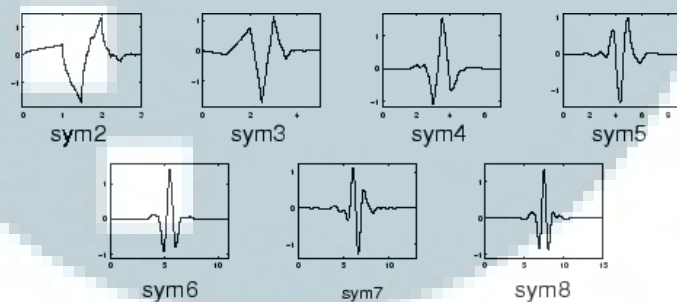
Gambar 2.9. *Wavelet* Daubechies [19]

Daubechies memiliki orde yang menggambarkan jumlah koefisien filternya seperti pada Gambar 2.9. Sifat polinomial yang dimiliki oleh *wavelet* akan berpengaruh dalam penentuan jumlah koefisien filter *wavelet*. Semakin besar jumlah filter yang dimiliki oleh suatu *wavelet* filter Daubechies, maka semakin baik filter tersebut dalam melakukan pemilihan frekuensi. Untuk

Daubechies orde  $N$  (db- $N$ ), maka Daubechies tersebut memiliki ukuran koefisien filter  $2N$ .

### 2.3.7. Wavelet Symlets

Wavelet Symlets memiliki nama pendek yaitu sym $N$  yang termasuk dalam keluarga Wavelet. Jenis ini merupakan hasil modifikasi dari wavelet Daubechies. Orde  $N$  dalam wavelet Symlets dituliskan dengan sym $N$  dan memiliki orde  $N=2, \dots, 45$ .  $N$  memiliki nilai integer positif. Panjang filter untuk wavelet Symlets adalah  $2N$ . Contohnya jika sym10 maka memiliki panjang filter yaitu 20. Gambar 2.10 menunjukkan jenis – jenis symlet.



Gambar 2.10. Wavelet Symlets [19]

## 2.4. Analisa Reduksi Noise

Analisa reduksi noise pada penelitian ini memiliki tiga aspek yang dapat diamati, yaitu analisa pengamatan objektif berupa *Mean Squared Error* (MSE), *Peak Signal-to-noise Ratio* (PSNR), dan pengamatan subjektif terhadap spektrum/sinyal.

### 2.4.1. Mean Squared Error (MSE)

*Mean Squared Error* (MSE) merupakan salah satu metode yang digunakan untuk menganalisis atau mengukur kesalahan sebuah metode. Dalam statistik, MSE sebuah estimator adalah nilai yang diharapkan dari kuadrat *error* yang menunjukkan seberapa besar perbedaan kualitas audio awal dengan kualitas audio setelah direduksi. Analisa ini dipilih karena memiliki nilai kesalahan yang cenderung kecil.

MSE merupakan cara pertama untuk mengukur kesalahan secara keseluruhan. Perbedaan tiap nilai MSE dapat terjadi karena adanya keacakan pada data atau karena estimator tidak mengandung informasi yang dapat menghasilkan estimasi yang lebih akurat. Rumus MSE adalah nilai *error* kuadrat rata – rata antara matriks asli dengan matriks hasil reduksi. Masing – masing dari hasil sisa dikuadratkan [20].

$$MSE = \frac{\sum_{t=1}^n (X_t - F_t)^2}{n} \quad (2.9)$$

Persamaan 2.9 merupakan rumus mencari nilai MSE. Variabel X merupakan nilai matriks audio aktual atau awal sebelum hasil reduksi, sedangkan F merupakan nilai matriks audio setelah direduksi dan N merupakan jumlah sampel atau matriks. Semakin kecil nilai MSE, maka metode reduksi yang digunakan semakin baik.

### 2.4.2. *Peak Signal-to-noise Ratio (PSNR)*

*Peak Signal-to-noise Ratio (PSNR)* adalah sebuah parameter yang biasa digunakan dalam reduksi *noise* untuk menentukan kualitas hasil audio setelah direduksi *noisenya*. PSNR membandingkan antara nilai maksimum dari sinyal yang diukur dengan besarnya *noise* yang berpengaruh pada sinyal tersebut. PSNR memiliki satuan desibel (dB). Fungsi dari analisa PSNR untuk mengetahui perbandingan kualitas audio sebelum dan sesudah ditambahkan *noise*, dikarenakan banyak sinyal yang mempunyai jarak dinamis (*dynamic range*) yang sangat lebar [20].

$$PSNR = 10 \log_{10} \left( \frac{C_{max}^2}{MSE} \right) \quad (2.10)$$

Rumus di atas merupakan rumus untuk mengetahui nilai PSNR.  $C_{max}^2$  memiliki nilai maksimum dari amplitudo audio. *Mean Squared Error (MSE)* juga ditentukan sebelumnya. Pada umumnya nilai PSNR disajikan dengan angka desimal yaitu dua angka dibelakang koma.

Nilai PSNR yang tinggi melambangkan reduksi *noise* yang besar antara sinyal sebelum dan sesudah disisipkan *noise*. Sedangkan nilai PSNR yang kecil melambangkan reduksi *noise* yang kecil antar sinyal sebelum dan sesudah ditambahkan *noise*.

### 2.4.3. Analisa Secara Subjektif

Analisa ini berdasarkan karakteristik visual manusia yang mampu melakukan pendeteksian secara langsung. Analisa ini menentukan baik buruknya spektrum hasil pengolahan yang ditentukan oleh pengamat sendiri

sehingga bisa terjadi dua buah sinyal yang sama pada kriteria objektif namun dapat berbeda kualitas subjektivitasnya, tergantung dari persepsi visual pengamat. Selain pengamatan juga dilakukan pengamatan subjektif terhadap kualitas suara melalui pendengaran.

## 2.5. Teori Musik

Musik adalah bunyi yang diatur menjadi susunan pola. Menurut Kamus Besar Bahasa Indonesia, Musik adalah nada atau suara yang disusun sedemikian rupa sehingga mengandung irama, lagu, dan keharmonisan (terutama yang menggunakan alat-alat yang dapat menghasilkan bunyi) [21]. Sedangkan menurut Kamus Merriam-Webster, musik adalah seni mengombinasikan nada-nada sedemikian rupa sehingga nada-nada itu menyenangkan, mengungkapkan perasaan atau dapat dimengerti [22]

Musik memiliki unsur – unsur yang membuatnya menjadi rangkaian melodi. Pada penelitian ini, komposer berhubungan dengan tempo, *beat*, *tone*, dan dinamika.

### 2.5.1. Tempo

Tempo adalah tingkatan kecepatan sebuah komposisi dimainkan dalam *beat* per menit [23]. Sedangkan menurut Kamus Musik [24], tempo berarti waktu, kecepatan dalam ukuran langkah tertentu. Dari kedua penjelasan tersebut dapat disimpulkan bahwa tempo berhubungan dengan cepat atau lambatnya sebuah komposisi dimainkan per menit. Tempo

menggunakan ukuran jumlah ketukan dalam satu menit, atau *beat per minute* (bpm) [25].

Terdapat beberapa macam tempo yang digunakan dalam musik, dikelompokkan menjadi Tempo Pelan (*Slow Tempos*), Tempo Sedang (*Moderat Tempos*), dan Tempo Cepat (*Fast Tempos*). Tempo Pelan atau *Slow Tempos* memiliki kecepatan antara 58 – 63 bpm. Tempo Sedang memiliki kecepatan antara 88 – 96 bpm, sedangkan Tempo Cepat memiliki kecepatan di atas 100 bpm [25].

	←---SLOWER			FASTER---→		
<i>Italian terms:</i>	<i>Adagio</i>	<i>Andante</i>	<i>Moderato</i>	<i>Allegro (assai)</i>	<i>Vivace</i>	<i>Presto</i>
<i>Beats/minute</i>	60	80	96	140	175	
	Blues	Pop	Funk	R & B	Rock and Roll	

Gambar 2.11. Tabel tempo dan nilai kecepatannya [26]

### 2.5.2. *Beat*

Musik telah terdengar dari waktu ke waktu, salah satu cara utama musik disederhanakan adalah dengan membagi keseluruhan musik menjadi periode yang kecil yang dinamakan *beat*. Pada musik umumnya *beat* terjadi tepat pada bagian awal, sehingga mudah untuk didengarkan dan dirasakan. Contohnya pada saat bertepuk tangan, menghentakkan jari-jari kaki, atau menari [27].

*Beat* dalam musik biasanya stabil, seperti detak jantung atau detik jam. Irama dalam karya musik berdasarkan *beat* yang konstan. Musik dapat ditulis dengan bpm yang berbeda-beda. Mayoritas lagu mempunyai empat *beat* pada satu birama [28]. Gambar 2.12. menunjukkan *beat* dalam birama 4/4.



Gambar 2.12. *Beat* dalam birama 4/4 [29]

## 2.6. Jenis Musik dalam Aplikasi Komposer

Perekaman suara dalam aplikasi komposer *beat* ini dapat dilakukan dengan berbagai macam tujuan atau jenis *beat* yang ingin dihasilkan. *User* harus mengetahui dan merancang konsep musik sehingga dapat menghasilkan *beat* yang diinginkan. Jenis musik yang cocok dalam penerapan di aplikasi ini antara lain *beatbox* dan musik eksperimental.

### 2.6.1. *Beatbox*

*Beatbox* adalah salah satu bentuk seni yang menggunakan organ vokal manusia untuk mengimitasi bunyi-bunyian ritmis dan ketukan drum, instrumen musik, maupun tiruan bunyi-bunyi lainnya Referensi mengenai *beatbox* pertama kali diterbitkan oleh Patryk Tiktak Matela pada tahun 2014 [30]. Dalam buku tersebut dijelaskan istilah “*Human Beatbox*” muncul pada



masa awal hip hop ketika pada *Disc Jockey* (DJ) menggunakan mesin drum elektronik untuk membentuk suara *bass* dan transisi antara lagu.

*Beatbox* berhubungan dengan *genre-genre* musik lainnya. Dalam referensi lain (Mickey Hess), *beatbox* merupakan istilah yang digunakan untuk seseorang yang menirukan ketukan drum dengan menggunakan mulut. Penerapannya dapat digunakan dalam musik hip hop [31]. Proses melakukan *beatbox* terjadi pada otot – otot yang sama yaitu di area mulut atau organ vokal. Namun terdapat perbedaan dalam hal cara memproduksinya. Terdapat 3 teknik dasar pengucapan dalam *beatbox* [1], yaitu

1. *Bass drum* atau *kick drum*. Teknik dasar ini dimainkan dengan cara seperti mengucapkan “dug”, tetapi ada juga yang memainkan dengan ucapan lain, misalnya “bwuh” atau “bub”.
2. *Hi-hat*. *Hi-hat* merupakan nama alat musik pada bagian drum, yang terbuat dari lempengan logam berjumlah 2 buah, ditumpuk menjadi 1 dan dimainkan secara kombinasi kaki kiri saat dipukul. Pada *beatbox*, *hi-hat* biasanya diistilahkan dengan huruf T dengan pengucapan seperti bunyi “cis” atau “ces”.
3. *Snare*. Teknik dasar ini menirukan atau mengimitasi suara *snare* pada drum. Pengucapan pada *beatbox* seperti bunyi “Kih”

### 2.6.2. Musik Eksperimental

Musik eksperimental merupakan bidang musik dimana komposisi musik yang biasanya ada, dikembangkan dan dilakukan eksplorasi yang lebih

jauh. John Cage adalah salah satu komposer paling awal dalam menciptakan musik eksperimental. Eksplorasi dalam bidang musik ini tidak mengenal batas. Beberapa komposer menggunakan musik ini berdasarkan hukum fisika dan ilmu pengetahuan alam lainnya. Musik ini umumnya menggunakan alat musik yang tidak biasa, seperti benda-benda di sekitar manusia. Improvisasi adalah kunci utama dalam memainkan musik eksperimental [32].

