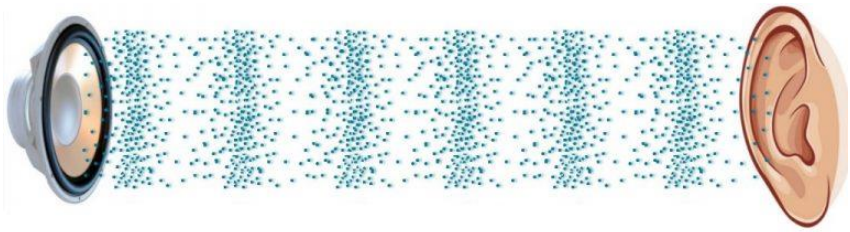
BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Suara

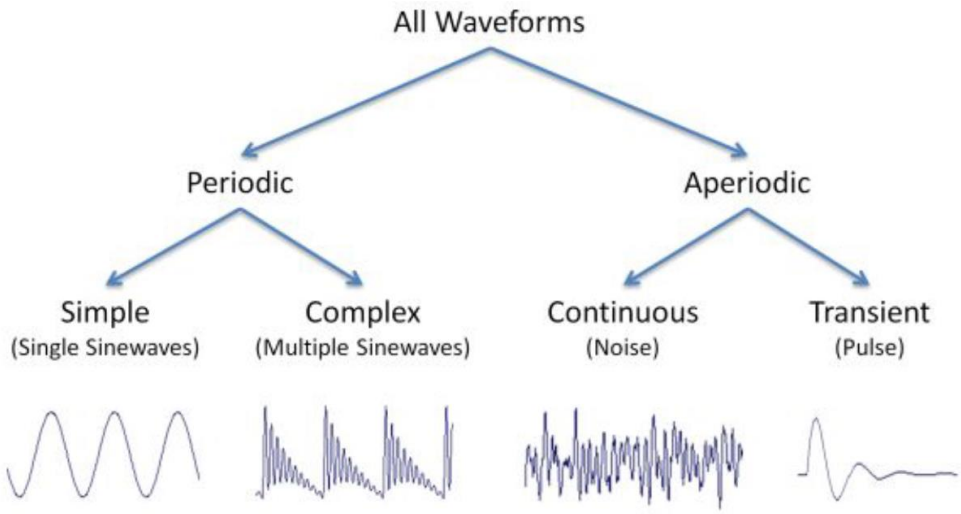
Suara adalah hasil vibrasi atau gesekan partikel objek dari suatu sumber baik dengan sifat medium udara, cairan, atau padat yang menyebabkan pelepasan energi ke berbagai arah sehingga menimbulkan perbedaan tekanan pada udara yang menciptakan gelombang. Suara termasuk gelombang mekanis dimana ketika pelepasan energi dari sumber ke arah lain menyebabkan partikel pada medium udara terkena pengaruh dan akan berkontraksi dan berelaksasi secara kontinu, dinamakan osilasi suatu gelombang (Müller, 2015), seperti yang terlihat pada gambar 2.1



Gambar 2.1 Ilustrasi bergeraknya suara
Sumber : (Ashish, 2018)

Suara lebih mudah melihat gambaran bentuk nya ketika di representasikan menggunakan bentuk gelombang yang paada dasarnya adalah plot grafik untuk meggambarkan tekanan yang dialami oleh udara seiring waktu,. Pada skema gelombang suara memberikan informasi seperti frekuensi, intensitas, dan timbre (Müller, 2015).

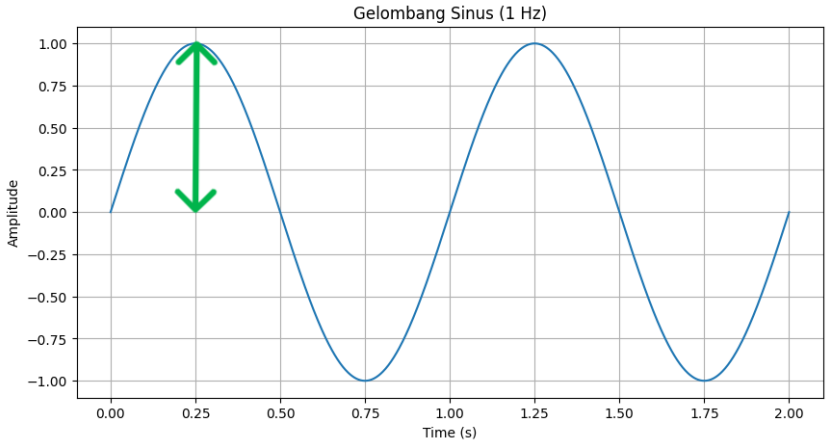
Gelombang suara memiliki dua kategori diantaranya periodik dan aperiodik. Gelombang suara periodik memiliki sifat kontraksi dan relaksasi tekanan kontinu yang terlihat jelas polanya, ada gelombang suara dengan pola sederhana salah satunya seperti gelombang sinus, adapun yang kompleks dengan beberapa gabungan gelombang sinus. Seperti namanya kebalikan dari periodik, gelombang suara aperiodik tidak terlihat jelas polanya, ada yang terlihat seperti nois atau gabungan dari beragam macam poin titik tekanan, adapun yang terlihat seperti ledakan tekanan mendadak pada satu titik perode saja layaknya sebuh detakan jantung. (Müller, 2015)



Gambar 2.2 Jenis bentuk gelombang suara
Sumber: (Velardo, 2020a)

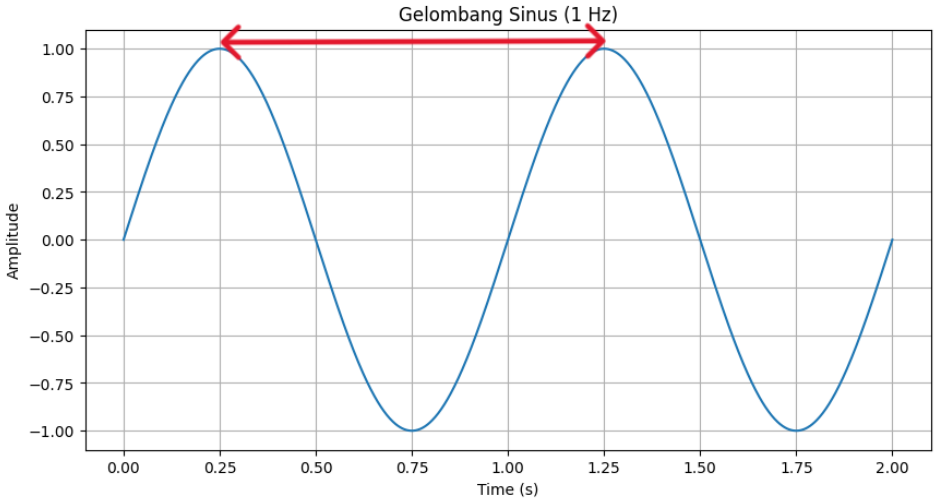
Contoh bentuk gelombang suara dengan rumus sinus sederhana ketika $y(t) = Asin(2\pi ft + \varphi)$ dimana A adalah amplitudo, f adalah frekuensi, t adalah waktu, dan φ adalah fase sinus (pada fase apa gelombang suara dimulai). Amplitudo adalah tingginya tekanan yang dialami ketika suara muncul, semakin

tinggi maka semakin nyaring digambarkan dengan garis garis panah hijau pada gambar 2.3 yang merupakan gambar grafik gelombang sinus dengan frekuensi 1 Hz.



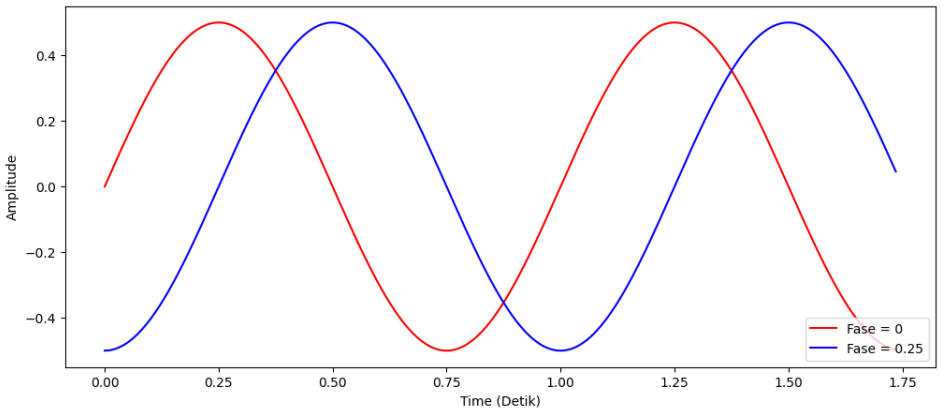
Gambar 2.3 Gambaran amplitudo pada gelombang sinus 1 Hz

Periode waktu pada konteks ini adalah waktu yang dibutuhkan untuk menggapai dua putaran puncak, rentang perode waktu digambarkan dengan garis merah pada gambar 2.4 grafik gelombang sinus.



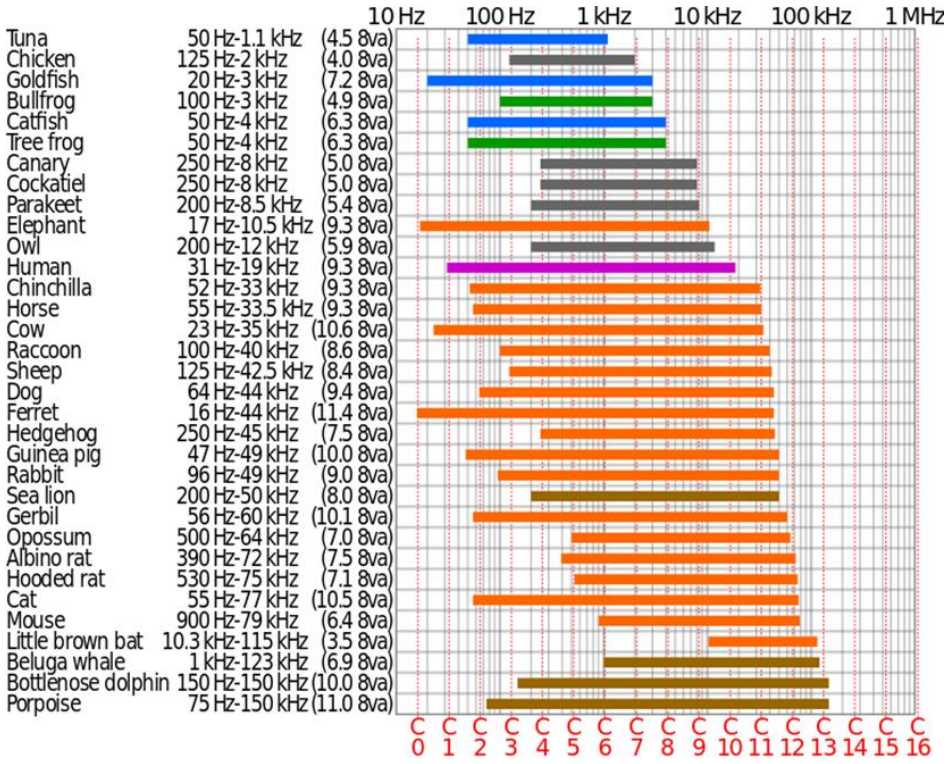
Gambar 2.4 Gambaran satu periode pada gelombang sinus dalam 1 Hz

Frekuensi adalah inversi dari perode, maka semakin tinggi frekuensi semakin tinggi juga nada suara yang didengar. Fase dilambangkan dengan phi (φ), sesuai dengan namanya fase berguna untuk menggeser gelombang ke kanan ketika φ bernilai positif atau kekiri ketika φ bernilai negatif. Grafik pergeseran diilustrasikan pada gambar 2.5 gelombang sinus dengan frekensi 1 Hz, ketika gelombang merah diterapkan perubahan fase sebesar 0.25, maka gelombang akan bergeser kekanan menjadi gelombang berwarna biru.



Gambar 2.5 Perbedaan bentuk gelombang suara ketika fase digeser sebesar 25%

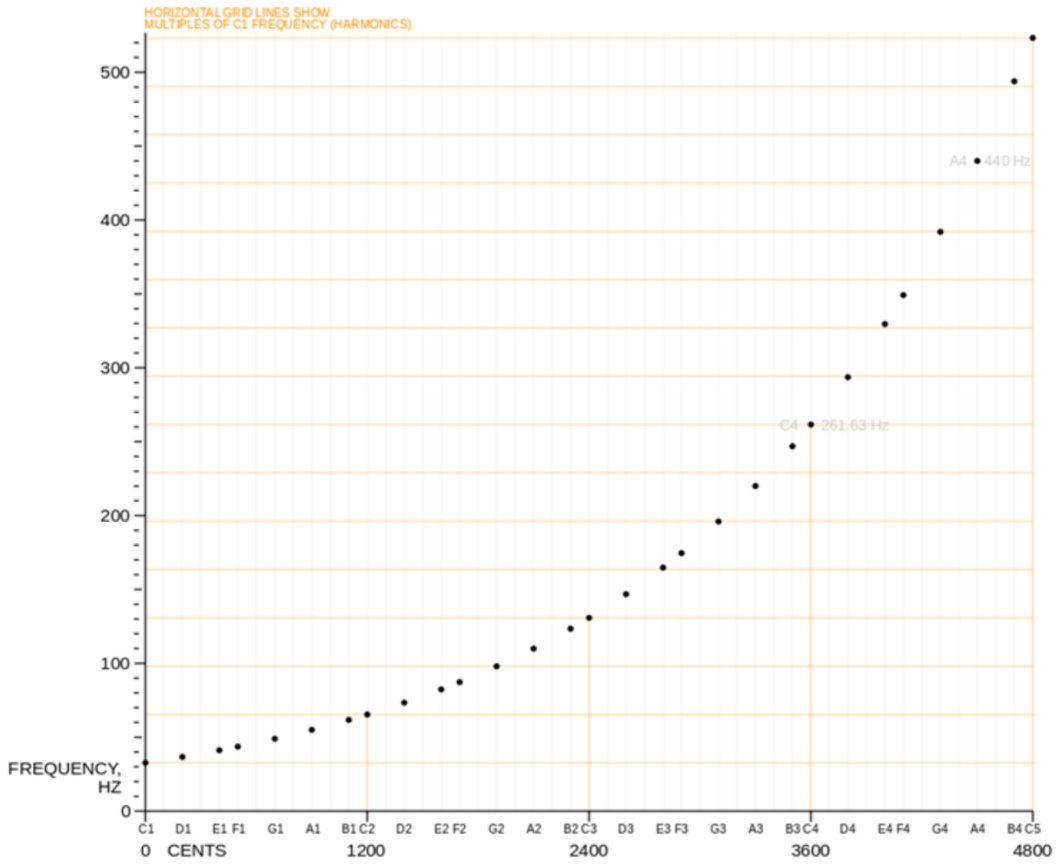
Jarak atau rentang pendengaran suara dapat didefinisikan sebagai jarak frekuensi yang bisa didengar oleh manusia dan makhluk hidup lainnya. Pada gambar 2.6 jarak pendengaran manusia secara umum berada pada rentang 20–20.000 Hz terkadang bervariasi pada setiap individu terutama pada frekuensi tinggi. Digambarkan juga perbandingan rentang persepsi frekuensi manusia dengan makhluk hewan lain seperti anjing dengan rentang 64–44.000 Hz, atau kucing dengan rentang 55–77.000 Hz (Baron, 1983; Richard R. Fay and Arthur N. Popper, 1994)



Gambar 2.6 Grafik logaritmik jarak pendengaran
Sumber: (Cmglee, 2014)

2.1.2 *Pitch* atau Tala

Pitch atau tala merupakan konsep perseptual yang memungkinkan suara direpresentasikan ke dalam skala yang berubungan dengan frekuensi terkait, jadi pendengar suara dapat menilah tinggi rendahnya melodi suara yang di dengarnya. Konsep ini menjelaskan bahwa manusia tidak mendengar suatu frekuensi secara linear tapi manusia mendengarnya secara logaritmik seperti yang terlihat pada gambar 2.7 (Stevens & Volkmann, 1940).



Gambar 2.7 Grafik kurva representasi tangga nada (*note*) terhadap frekuensi
Sumber: (Bill, 2006)

Pitch memiliki hubungan erat dengan frekuensi, namun keduanya bukanlah hal yang sama, karena frekuensi merupakan satuan ilmiah yang dapat diukur, sedangkan *pitch* adalah sebuah konsep perseptual manusia terhadap gelombang suara tergantung individu yang mendengarnya, meskipun begitu sebagian besar kelompok individu setuju terhadap skala nada musik mengenai nada mana yang lebih tinggi dan lebih rendah. *Pitch* membuat manusia bisa menetapkan sensasi suara yang didengarnya pada skala musik menggunakan nada musik (Popper, Arthur N, Simmons, Andrea M, Fay, 2003).

2.1.3 *Loudness* (Kenyaringan)

Pada sisi audio, variasi terhadap tinggi rendahnya suara yang diterima alat pendengaran biasa disebut dengan kebisingan, kenyaringan atau *loudness*, dimana kerasnya suara ini dapat diukur pada skala dari suara pelan hingga keras. Kebisingan merupakan satuan subjektif yang berhubungan dengan intensitas suara dan kekuatan suara, karakteristiknya juga tidak luput dari kolerasinya dengan durasi dan frekuensinya (Müller, 2015).

Tidak mudah menentukan dengan pasti intensitas atau kekuatan suara secara fisik. Umumnya daya/tenaga dalam bahasa inggris yaitu *power* menjadi acuan pengukuran intensitas suara dengan satuan watt (W) yang didefinisikan sebagai satu joule per detik. Konsep daya sama seperti daya pada energi listrik semakin besar watt semakin besar pula energinya, daya pada intensitas suara merepresentasikan energi per waktu yang dipancarkan dari sumber suara ke segala arah melewati udara. Intensitas suara kemudian dinyatakan dalam daya/tenaga per satuan unit area (Müller, 2015).

Kenyataanya, intensitas suara yang sangat kecil masih bisa didengar oleh manusia, contohnya *threshold of hearing* (TOH) atau ambang batas pendengaran minimum murni ini memiliki besaran $I_{TOH} := 10^{-12} W/m^2$. Tidak hanya itu, rentang pendengaran manusia terhadap intensitas suara yang dirasakan bisa sangat besar hingga $I_{TOP} := 10\, W/m^2$ juga sebagai ambang batas pendengaran merasakan kesakitan atau istilahnya *Threshold of Pain* (TOP). Dalam praktisnya untuk mengukur daya dan intensitas suara berali ke skala desibel (dB) yang bersifat logaritmik sehingga pengukurannya menggunakan dua unit, unit pertama yaitu

I_{TOH} digunakan sebagai referensi atau patokan intensitas suara, dan besaran intensitasnya diukur dalam dB yang ditetapkan dengan persamaan 2.1

$$dB(I) := 10 \cdot log_{10}\left(\frac{I}{I_{TOH}}\right) \quad (2.1)$$

Jadi, untuk besaran intensitas $dB(I_{TOH}) = 0$, dan duakali lipatnya intensitas suara menghasilkan besaran intensitas yang menaik sekitar 3 dB:

$$dB(2 \times I) := 10 \cdot log_{10}(2) + dB(I) \approx 3 + dB(I) \quad (2.2)$$

Level intensitas menunjukkan nilai intensitas dalam W/m^2 dan besaran intensitas dalam dB kedalam beberapa kelas sumber suara dan indikator ditunjukkan pada tabel 2.1

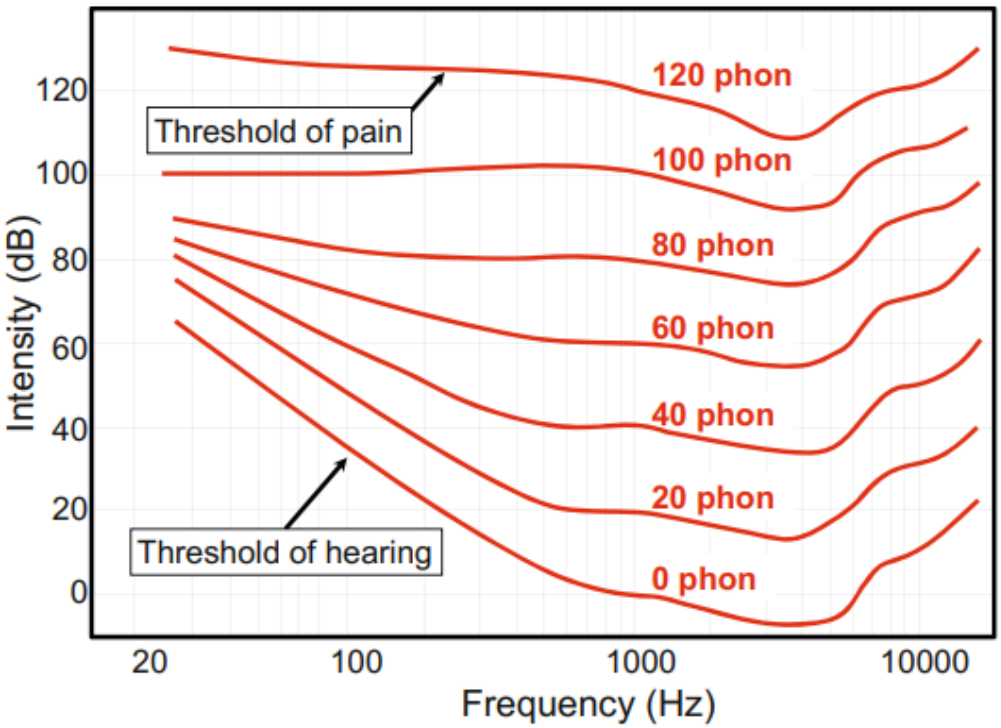
Tabel 2.1 Level Intensitas Suara
Sumber: (Müller, 2015)

Sumber	**Intensitas**	**Besaran Intensitas**	**× TOH**
Threshold of Hearing (TOH)	10^{-12}	0 dB	1
Bisikan	10^{-10}	20 dB	10^2
Pianissimo	10^{-8}	40 dB	10^4
Percakapan biasa	10^{-6}	60 dB	10^6
Fortissimo	10^{-2}	100 dB	10^{10}
Threshold of Pain (TOP)	10	130 dB	10^{13}
Jet lepas lndas	10^2	140 dB	10^{14}
Kerusakan instan gendang telinga	10^4	160 dB	10^{16}

Kebisingan suara dipengaruhi berbagai faktor. Suara yang sama akan memiliki kebisingan yang berbeda saat dirasakan oleh orang yang berbeda, hal ini disebabkan oleh beberapa faktor seperti usia atau kondisi alat pendengaran. Durasi suara juga mempengaruhi persepsi manusia terhadap suara tersebut, karena sistem pendengaran manusia mengambil efek rata–rata dari intensitas suara dalam interval satu detik. Disebabkan oleh hal ini, manusia merasakan bahwa suara berdurasi 200 ms lebih bising dibandingkan dengan suara yang sama namun berdurasi 50 ms,

untuk kedua suara yang memiliki intensitas yang sama tetapi berbeda frekuensi pun tidak dirasa oleh manusia memiliki tingkat kebisingan yang sama.

Berdasarkan dari eksperimen psikoakustik (Müller, 2015), kebisingan nada murni yang dirasakan manusia berdasarkan frekuensinya telah ditentukan dan dinyatakan dalam satuan unit phon ditujukkan pada gambar 2.8 sebagai kontur persamaan kebisingan (*equal loudness contours*).



Gambar 2.8 Kontur Persamaan Kebisingan
Sumber: (Müller, 2015)

Setiap garis kontur menentukan kebisingan tetap dalam satuan phon sebagai intensitas suara terhadap sumbu frekuensi. Satuan unit phon mulai dinormalisasi ketika frekuensi 1000 Hz, dimana nilai phon setara dengan besaran intensitas dalam dB.

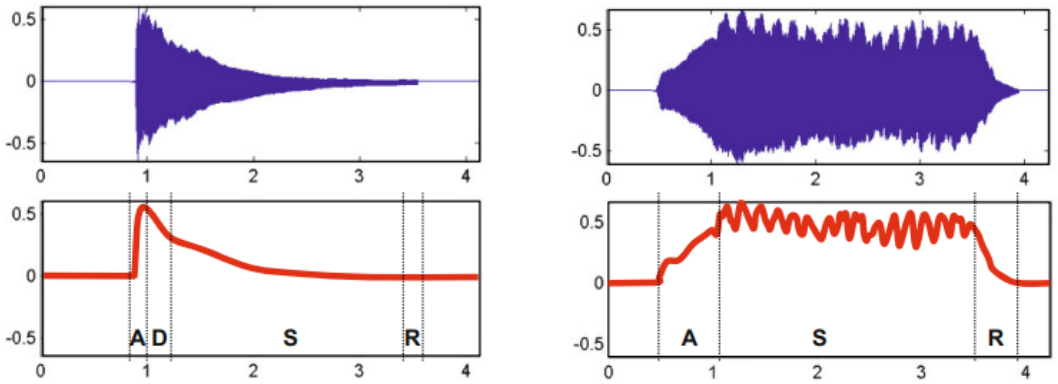
2.1.4 Timbre

Persepsi timbre oleh para penekun bidang musik merupakan warna dari suara. Timbre masih sulit dijelaskan karena fenomena bentuk multidimensionalnya yang sulit diukur, biasanya sering dirasakan perbedaannya secara tidak langsung contohnya seperti merasakan perbedaan dari kedua sumber suara meskipun memiliki intensitas, frekuensi, dan durasi yang sama pada suara A4 biola dan piano. Dalam timbre suara sering dideskrispikan dengan kata seperti cerah, kelam, redup, kasar, hangat, dan sebagainya.

Peneliti berusaha untuk mencari pendekatan terbaik dalam menggambarkan timbre dengan melihat kolerasinya dengan karakteristik suara seperti ada tidaknya tonal, perkembangan spektral dan temporalnya, komponen noise, atau distribusi energi nada, beberapa sifat timbre diantaranya envelope suara, konten harmoniknya, dan modulasi frekuensi/amplitudo.

Ketika menekan tuts piano, suara yang dihasilkan mengandung gabungan suara kompleks dengan karakteristik yang berubah seiring waktu, dilihat pada bagan grafis membentuk skema gelombang *envelope* yang membentuk kurva halus. Perbedaan pada kurva halus ini dapat dijelaskan dalam model yang bernama ADSR. ADSR singkatan dari fase *Attack, Decay, Sustain,* dan *release*. Fase *attack* yang mengandung ledakan energi mendadak ketika menekan tuts piano bersamaan dengan noise yang disebabkan hasil dari mekanisme piano, fase *decay* ketika suara mulai stabil, fase *sustain* dimana energi pada suara mulai konstan, dan fase *release* ketika suara mulai memudar. Durasi untuk setiap fase yang bersangkutan sangat mempengaruhi bentuk suara yang keluar dan didengan manusia.

Gambar 2.9 memberikan ilustrasi penerapan model ADSR kepada suara alat musik piano dan biola. Bentuk envelope amplitude ketika memainkan nada C4 di piano serupa dengan ilustrasi model ADSR, ada peningkatan tajam energi dalam amplitudo ketika fase *attack* disaat pencetik dalam mekanik piano memukul senar, kemudian fase *decay* hingga energi suara mulai stabil hingga suaranya memudar. Bentuk envelope amplitude pada biola terlihat berbeda, peningkatan bertahap amplitude pada fase *attack*, tidak ada fase *decay* dan fase *sustain* yang tidak stabil, tetapi berosilasi secara teratur layaknya bagaimana memainkan senar pada biola, pada fase *release* terjadi ketika busur berhenti berkesekan dengan senarnya, suara biola dengan cepat memudar.



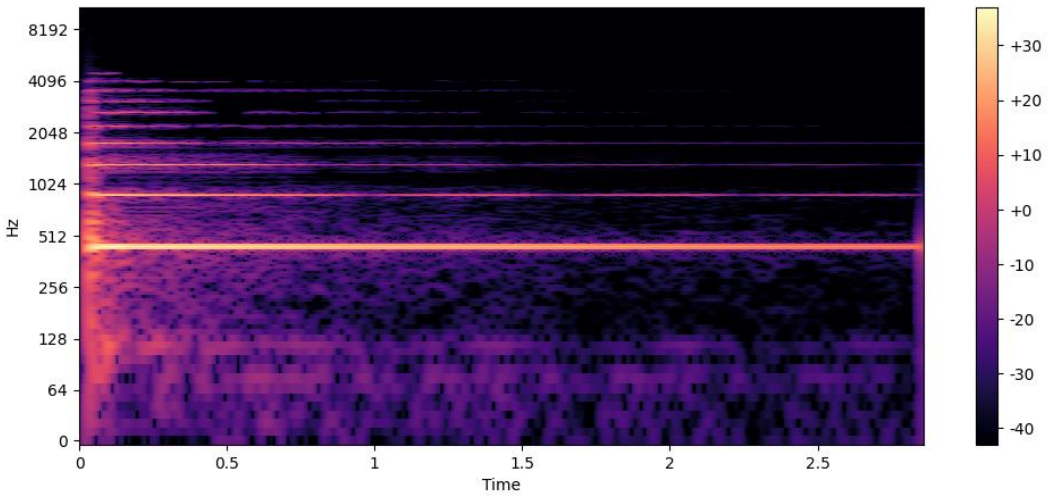
Gambar 2.9 Gelombang suara dan envelope amplitudo nada C4 Piano (kiri) dan biola (kanan)
Sumber: (Müller, 2015)

Suara yang timbul disekitar manusia bukan hanya sekedar nada yang timbul dengan frekuensi sederhana. Ketika memainkan satu nada pada sebuah instrumen akan menghasilkan nada kompleks yang mengandung berbagai campuran frekuensi berbeda seiring berjalannya waktu. Nada kompleks musikal seperti ini dapat disebut sebagai superposisi dari nada murni atau sinusoid, masing–masing dengan frekuensi, amplidue dan fase mereka sendiri. Parsial merupakan salah satu sinusoid

dari nada yang timbul dari nada musikal. Parsial dengan frekuensi terendah dinamakan frekuensi dasar dari suara. Pitch nada musikal biasanya ditentukan dari nilai frekuensi dasarnya. Harmonik parsial adalah sebuah parsial yang memiliki frekuensi sebesar multiplikatif dari frekuensi fudamentalnya, terhitung naik dari frekuensi fudamental. Spektogram nada piano A4 gambar 2.10 menunjukkan harmonik parsial dari frekuensi fudamentalnya yaitu 440 Hz terlihat dari heatmap yang paling terang dan paling bawah. Parsial selanjutnya dapat dihitung dengan mengkalikan fudamental dengan integer seperti pada persamaan 2.3

$$f_1 = 440, f_2 = 2 \times 440 = 880, f_3 = 3 \times 440 = 1320, \ldots\ldots, f_n = n \times 440 \quad (2.3)$$

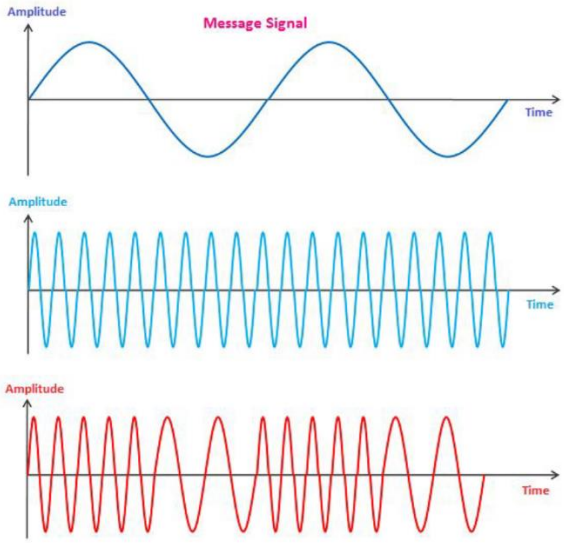
Istilah inharmonisitas mengidikasikan penyimpangan dari harmonik parsialnya, semisalnya memiliki frekuensi yang tidak termasuk bagian parsial dari frekuensi parsial.



Gambar 2.10 Spektogram Piano A4

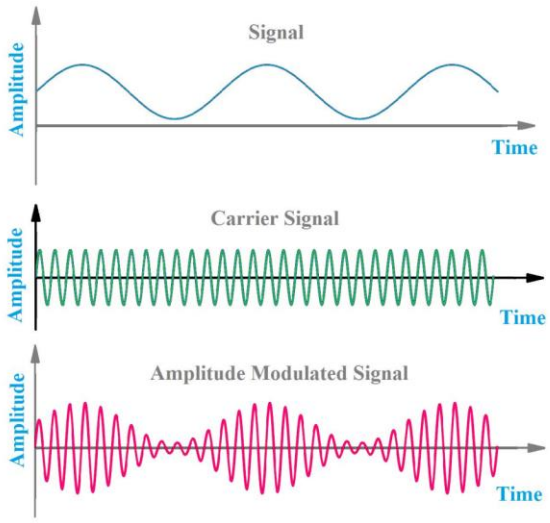
Modulasi frekuensi disebut juga vibrato oleh para musisi biasa digunakan untuk kepentingan ekspresif, memiliki variasi frekuensi secara periodik. Modulasi amplitude disebut juga tremolo memiliki fungsi dan cara kerja serupa dengan

vibrato tetapi amplitude nya yang diubah. Gambar 2.11 memberikan ilustrasi bentuk vibrato .



Gambar 2.11 Modulasi Frekuensi
Sumber: (Velardo, 2020b)

Pada gambar 2.12 berikut memperlihatkan bentuk sinyal modulasi amplitudo dengan efek sinyal tremolo pada sumber sinyal (*Carried Signal*)

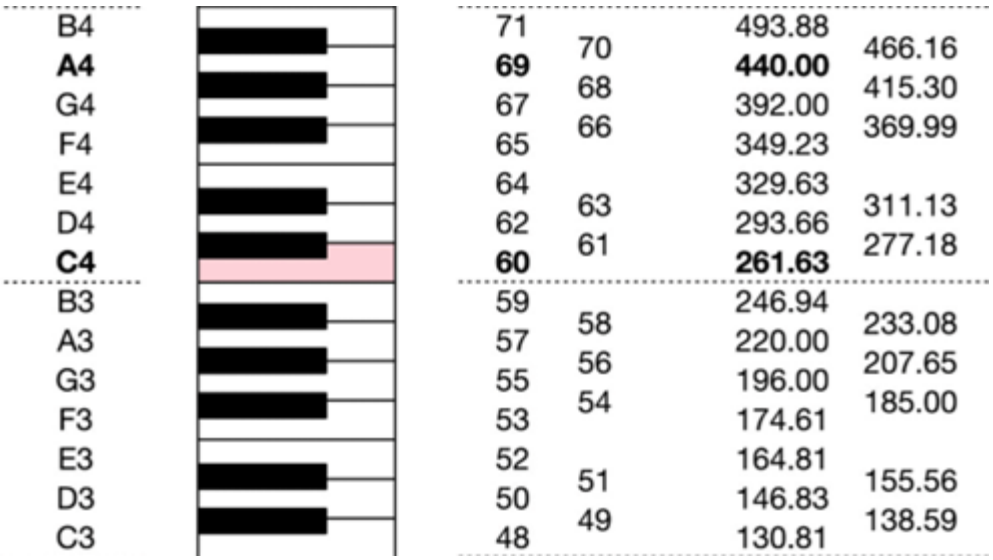


Gambar 2.12 Modulasi Amplitude
Sumber: (Velardo, 2020b)

2.1.5 MIDI

MIDI singkatan dari *Musical Instrument Digital Interface* merupakan standar internasional kode musik untuk perangkat keras dan perangkat lunak musik eletronik dan komputer (Müller, 2015). Nomor MIDI melingkup integer dari 0 hingga 127 dalam mengkode nada pitch.

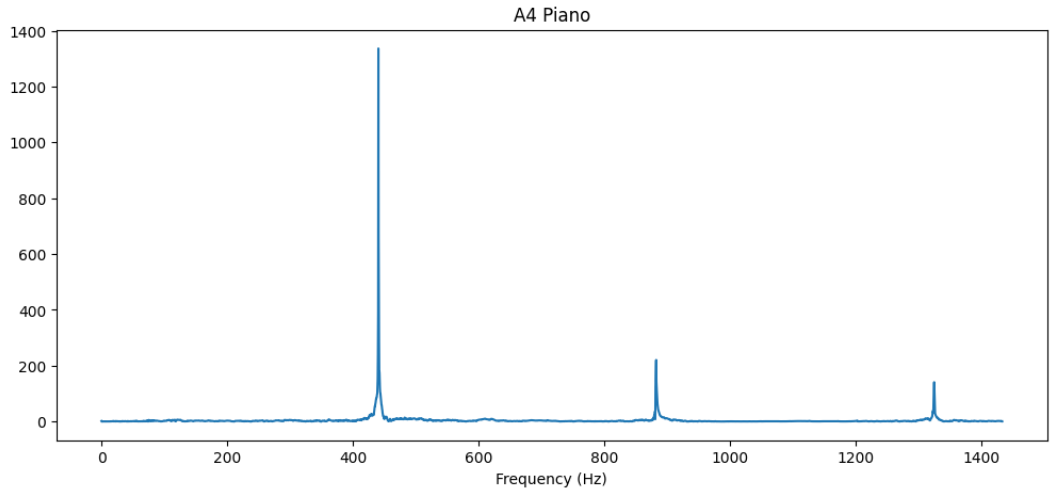
Gambar 2.13 sebagai gambaran dan bagan nada, nomor MIDI, dan frekuensinya. Dari nada bawah A0 atau nomor MIDI 21 hingga nada teratas C8 atau nomor MIDI 108 pada bagan nada MIDI gambar 2.13. Pola nama nada ditandai dengan tulisan afabetik contohnya pada C4, D4 E4, F4, G4, A4, dan B4, kemudian nomor setelahnya pada nama nada merupakan nilai *octave* atau oktan.



Gambar 2.13 Bagan Nada MIDI (Dibuat oleh Matthijs Hollemans)
Sumber: (Matthijs Hollemans, 2023)

Konsep satu oktaf yaitu ketika kedua frekuensi akan terdengar serupa atau harmonik ketika perbedaan frekuensinya kelipatan duanya, contohnya pada gambar 2.14, kunci nada MIDI A4 dengan frekuensi 440 Hz akan terdengar serupa dengan MIDI A5 dengan frekuensi 880 Hz namun terdengar lebih tinggi, dan juga

sebaliknya ketika dibandingkan dengan nada A3 dengan frekuensi 220 Hz akan terdengar lebih rendah dibanding dengan A4. Fenomena ini suka disebut dengan keajaiban dalam musik. Buktinya dapat dilihat dalam grafik frekuensi piano A4 pada gambar 2.14. kandungan frekuensi paling besar terdapat pada nada piano A4 tentu pada 440 Hz, namun terlihat ada sedikin puncak pada frekuensi 880 Hz, frekuensi ini adalah oktaf dan harmonik parsial dari nada A4, kemudian ada lagi sedikit puncak pada frekuensi 1320 Hz yang juga merupakan harmonik parsial dari A4.



Gambar 2.14 Grafik Plot Frekuensi Piano A4

Mapping pitch ke frekuensi bisa dilakukan dengan persamaan 2.4. Dimana F adalah frekuensi dalam Hz, p adalah nomor pitch dalam nada MIDI, misalnya untuk nomor nada MIDI 69 memiliki frekuensi sebesar 440 Hz.

$$F(p) = 2^{\frac{p-69}{12}} \cdot 440 \qquad (2.4)$$

Sebaliknya mapping frekuensi ke pitch dapat dilakukan dengan persamaan 2.5, dimana F adalah frekuensi dalam Hz dan p adalah nomor pitch dalam nada

MIDI

$$p = 69 + 12 \cdot log_2\left(\frac{f}{440\ Hz}\right) \quad (2.5)$$

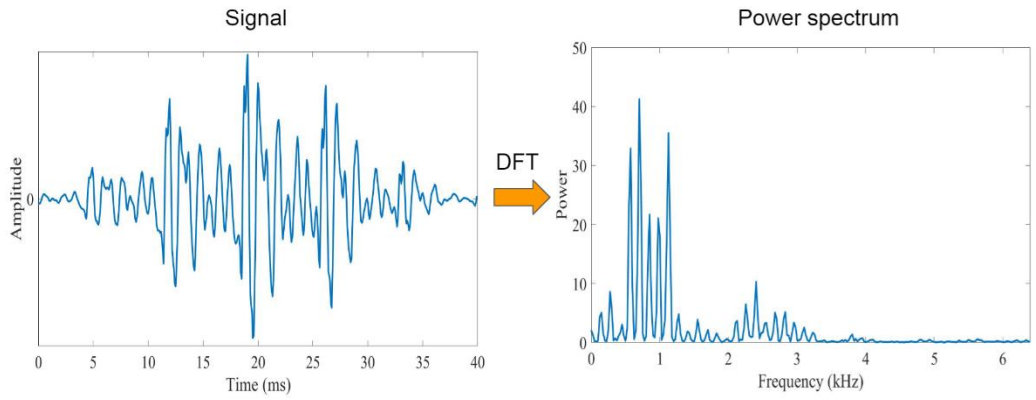
2.1.6 *Mel-Frequency Cepstrum Coefficients* (MFCC)

Mel-Frequency Cepstrum Coefficients merupakan koefisien kolektif dari hasil invers transformasi kosinus diskrit (*Discrete Cosine Transform*) kumpulan logaritma skala Mel yang secara persepsual merupakan skala relevan untuk menilai pitch dan menggambarkan karakteristik bagian suara.

Hasil dari ekstraksi dapat digunakan sebagai identitas objek sumber suara atau identitas seseorang (Marlina dkk., 2018). Metode ini sering digunakan karena sifat hasil ekstraksi fitur serupa dengan bagaimana manusia mendengar suara, dimana manusia paling sensitif ketika mendengar suara pada frekuensi setelah 1.000 Hz sesuai dengan grafil *equal loudness contours* pada gambar 2.7 (Sharif dkk., 2023).

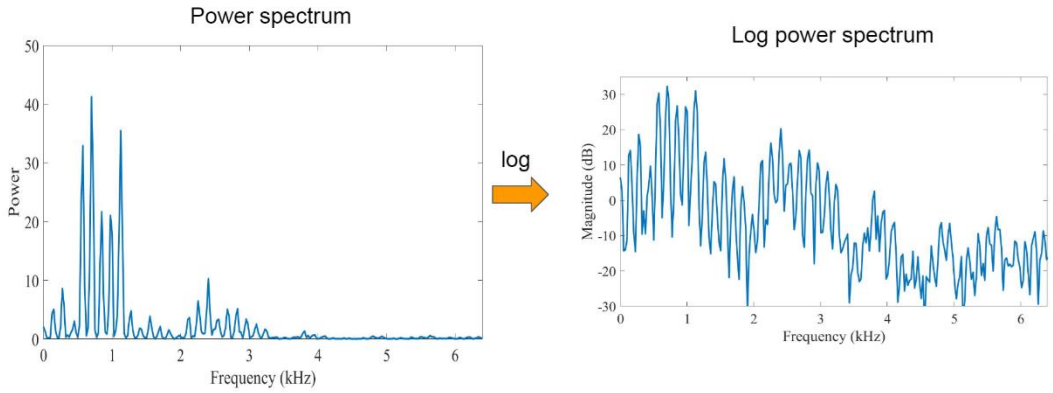
Visualisasi kasar dalam mencari cepstrum (parafrase dari *Spectrum*) yang merupakan spektrum dari log spektrum ditampilkan pada gambar 2.15 hingga gambar 2.17.

Dimulai dari gambar 2.15 tahap pertama mendapatkan gelombang suara diskrit semisal berdurasi 40 ms, lalu diubah dengan DFT (*Discrete Fourier transform*) untuk mendapatkan *power* spektrum dalam frekuensi (Hz).



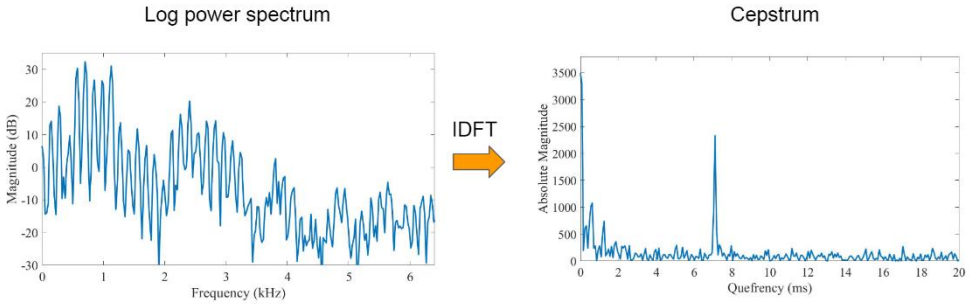
Gambar 2.15 Transformasi Gelombang Suara ke Plot Grafik frekuensi
Sumber: (Velardo, 2020c)

Tahap kedua pada gambar 2.16 melakukan transformasi dengan menerapkan logaritma untuk mendapatkan log spektrum ferkuensi terhadap besaran kebisingan dalam (dB).



Gambar 2.16 Transformasi Plot Grafik Frekuensi ke Spektrum Frekuensi
Sumber: (Velardo, 2020c)

Tahap ketiga pada gambar 2.17 melakukan *invers fourier transform* pada *log power spectrum* yang berbentuk seperti sinyal kontinu layaknya gelombang suara untuk mendapatkan spektrum dari spektrum gelombang suara, atau memiliki nama lain yaitu cepstrum yang memiliki satuan pseudo frekuensi atau *quefrency* (ms, parafrase dari frekuensi), pseudo frekuensi ini (Oppenheim & Schafer, 2004).

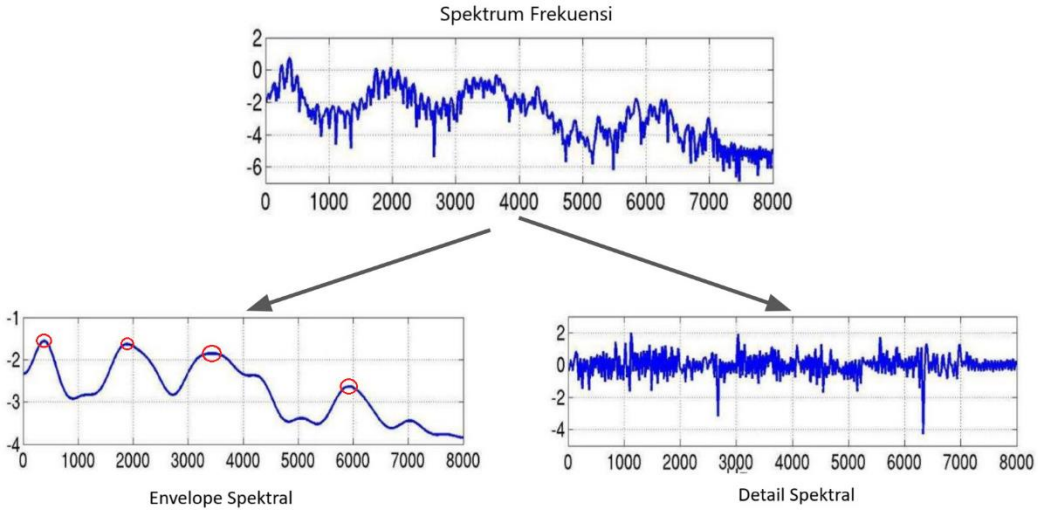


Gambar 2.17 Transformasi Spektrum Frekuensi ke Cepstrum
Sumber: (Velardo, 2020c)

Cepstrum memiliki keterkaitan dengan bagaimana manuisa menghasilkan suara untuk berbicara (Childers dkk., 1977) dengan memanfaatkan saluran vokalnya diilustrasikan dengan gambar 2.18. Diawali dengan glotis menghasilkan bunyi lalu melewati saluran suara. Saluran vokal memiliki mekanik kompleks dengan berbagai elemen yang berkontribusi dalam menghasilkan suara yang berbeda tergantung bagaimana elemen terkait digunakan seperti posisi lidah dalam menghasilkan suara vokal atau konsonan, sehingga dapat dikatakan bahwa saluran vokal bekerja sebagai filter suara yang bersumber dari glotis.

Gambar 2.18 Skema Manusia Menghasilkan Suara Vokal
Sumber: (Velardo, 2020c)

Suara vokal dapat diinterpentasikan sebagai konvolusi dari respon frekuensi dari saluran vokal dengan getaral glotis. Spektrum seperti pada gambar 2.19 di dekomposisikan menjadi envelope spektral dan detail spektral, envelope spektral menghaluskan grafik pada gelombang logaritma spektrum. Terlihat empat puncak gelombang dari envelope spektral pada gambar Gambar 2.19, puncak–puncak ini membawa timbre atau informasi atau identitas mengenai variasi fonem ketika manusia berbicara, bentuk envelope spektral serupa dengan respon frekuensi dari saluran vokal. ketika memisahkan spektrum dengan envelope spektral, akan menghasilkan detail spektral yang bentuknya dapat dipetakan dengan baik dengan getaran glotis (Norton & Karczub, 2003).



Gambar 2.19 Dekomposisi Spektrum Frekuensi
Sumber; (Velardo, 2020c)

Dekomposisi spektrum dapat ditulis dengan persamaan 2.6. Dimana x(t) adalah gelombang suara dalam frekuensi, e(t) getaran glotis, dan h(t) saluran vokal,

$$x(t) = e(t) \times h(t) \qquad (2.6)$$

Ketika diubah kedalam domain frekuensi akan menjadi persamaan 2.8

$$X(t) = E(t) \times H(t) \tag{2.7}$$

Lalu diterapkan logaritma pada kedua sisi persamaan akan menjadi persamaan 2.9

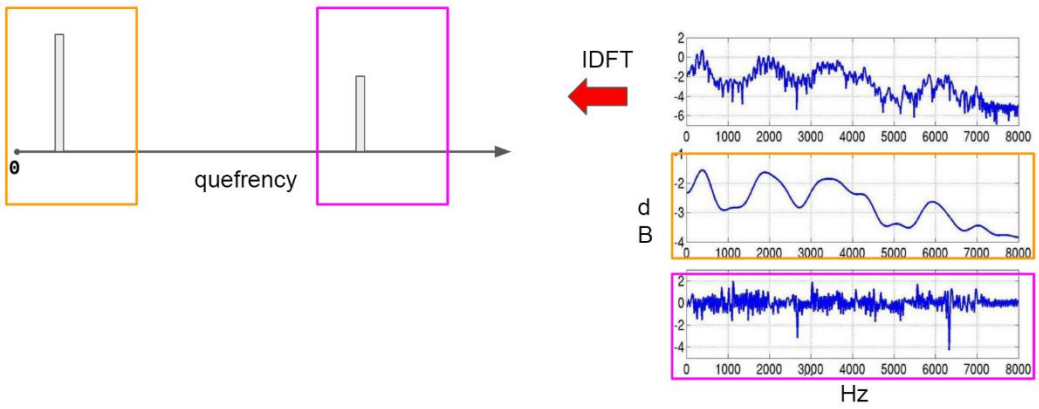
$$\text{Log}\big(X(t)\big) = \log\big(E(t) \times H(t)\big) \tag{2.8}$$

Atau dengan memanfaatkan sifat operasi aritmetika logaritma dapat ditulis menjadi persamaan 2.9

$$\text{Log}\big(X(t)\big) = \log\big(E(t)\big) + \log\big(H(t)\big) \tag{2.9}$$

Dalam bentuk persamaan 2.9 dimana $\log\big(X(t)\big)$ sama dengan spektrum frekuensi, $\log\big(E(t)\big)$ envelope spektral, dan $\log(H(t))$ detail spektral.

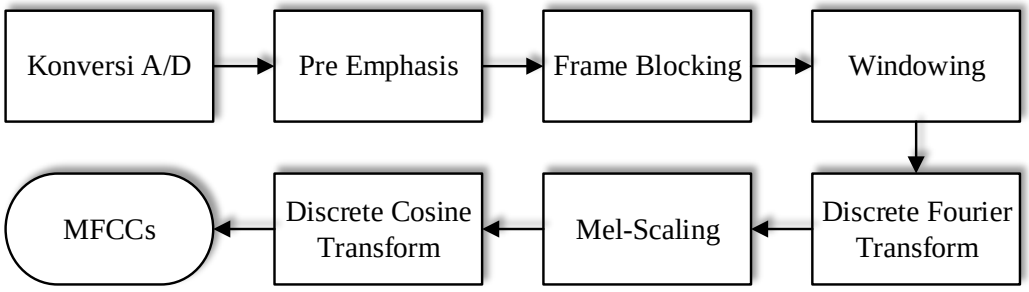
Perihal pemrosesan suara lebih berfokus pada identitas suara sehingga persamaan 2.9 sangat membantu karena bisa berfokus pada $H(t)$. cara mendapatkan $H(t)$ dengan menerapkan invers DFT terhadap spektrum frekuensi $\log\big(X(t)\big)$. Melihat konsep menggambarkan gelombang suara dalam plot, semakin tinggi frekuensi maka semakin kecil periode waktunya, maka gelombang semakin menyempit. Jika diperhatikan, envelope spektral dan detail spektral memiliki bentuk serupa dengan gelombang suara. Maka ketika melihat puncak quefrency pada plot grafik cepstrum mewakili envelope spektral dan detail spektral. Sehingga semua informasi pada cepstral dapat ditangkap melaui formalisasi matematis seperti pada persamaan 2.9



Gambar 2.20 Skema Cepstrum (H(t) daerah kuning, E(t) daerah ungu)
Sumber: (Velardo, 2020c)

Low pass liftering (parafrase dari filtering) dapat membatu dalam mendapatkan hanya dengan envelope spektral yang dalam kontek MFCC merupakan fitur suara lalu menghiraukan nilai dari detail spektrum, sesuai dengan namanya yaitu memfilter atau dalam bahasa ini menglifter bagian rendahnya quefrensi yang mengandung envelope spektral pada cepstrum.

Secara garis besar (Sharif dkk., 2023) , untuk mendapatkan MFCC hampir serupa dengan langkah mendapatkan cepstrum, langkahnya dapat digambarkan pada gambar 2.21.

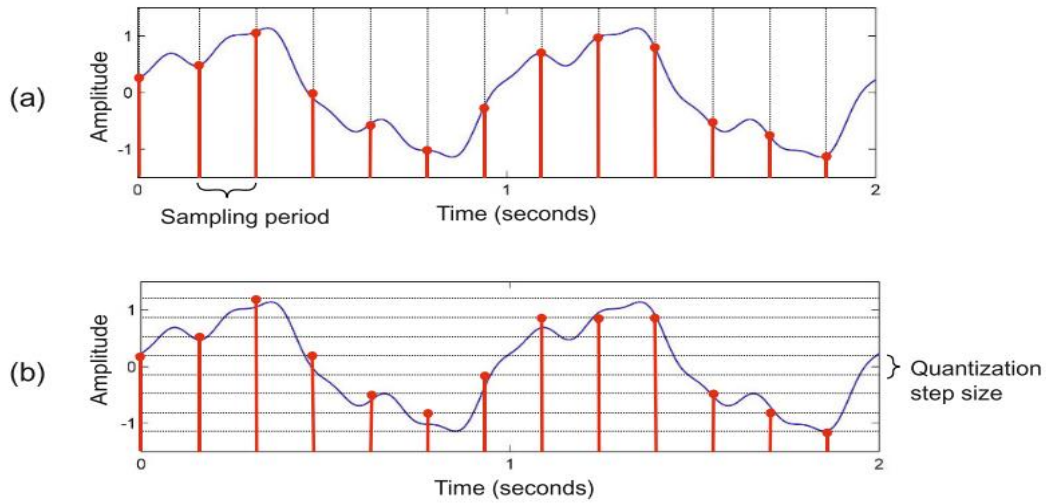


Gambar 2.21 Tahapan MFCC

a. Konversi A/D

Singkatan dalam bahasa inggris dari *Analog-to-Digital Conversion* merupakan sistem yang mengurangi kandungan sinyal analog yang kontinu ke

diskrit, cara umum mendigitalisasi audio biasa dilakukan dengan dua langkah yaitu sampling dan kuantisasi seperti pada gambar 2.22.



Gambar 2.22 Langkah konversi sinyal suara analog ke digital,
dimana (a) Sampling dan (b) kuantisasi
Sumber: (Müller, 2015)

Sampling audio analog kedalam poin - poin data dalam setiap periode waktu T tertentu seperti yang diilustrasikan pada gambar 2.22. Sampling dapat dimodelkan dengan persamaan 2.10

$$x(n) := f(n \cdot T) \tag{2.10}$$

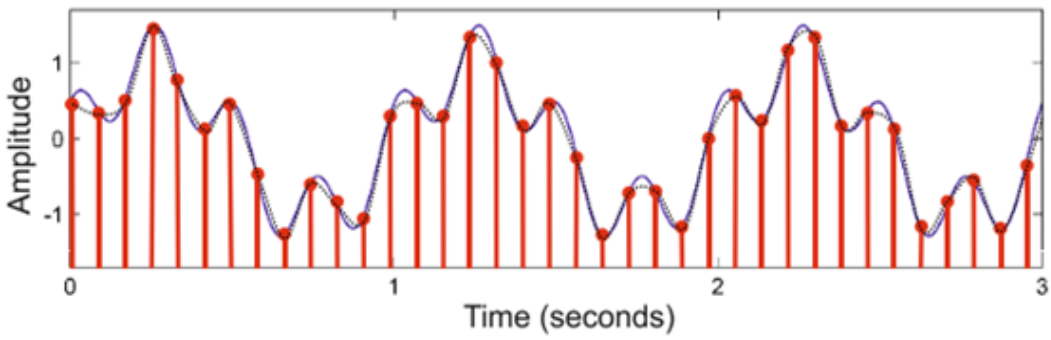
Dimana n merupakan bilangan integer, $x(n)$ merupakan sample yang diambil dlam waktu $t = n.T$ dari sumber audio analog. Proses ini dinamakan T-sampling. Nilai T adalah periode sampling dan nilai invers $F_s := 1/T$ dinamakan sampling rate. Sampling rate menandakan banyaknya sample yang ada dalam waktu satu detik, sampling rate memiliki satuan Hertz (Hz).

Langkah selanjutnya melakukan kuantisasi atau *quantization*. Mengganti amplitudo kontinyu dalam bilangan real dengan digit tak hingga ke dalam

amplitudo dalam bilangan real dengan digit tertentu. Contoh implementasi ada pada penyimpanan kaset seperti CD, terlihat pada label tulisan encoding seperti 8-bit atau 16-bit, yang artinya tingkat detail

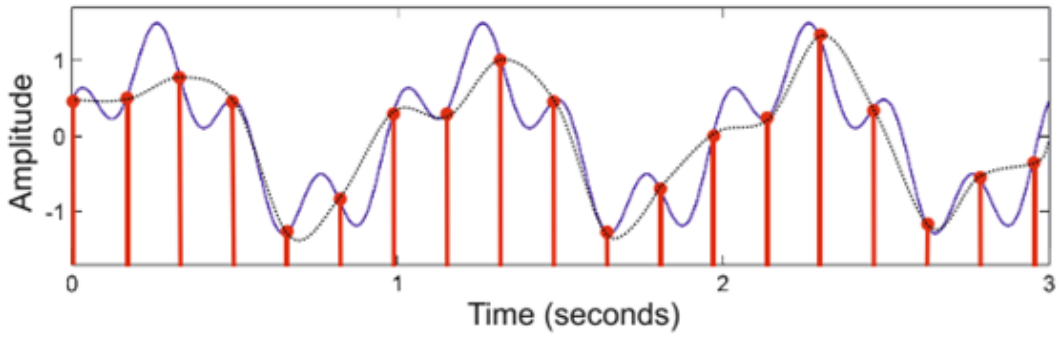
Umunya, dikatakan dalam teorema sampling, pada proses sampling akan kehilangan sebagian informasi. Audio asli dapat dibentuk kembali dari sample $x(n)$ jika audio original f tidak memiliki informasi frekuensi diatas $\Omega := F_s/2 = 1/(2T)$ Hz. Dapat dikatakan juga f adalah sinyal bandlimited, dimana Ω dikenal sebagai frekuensi Nyquist. Bilamana f mengandung frekuensi tinggi tersebut, maka sampling dapat memiliki kecacatan yang dinamakan aliasing seperti yang diilustrasikan pada gambar 2.23 hingga gambar 2.5.

Bentuk sinusoid audio pada gambar 2.23 hingga gambar 2.25 sangat berbeda ketika melakukan sampel gambar 2.23 dengan frekuensi 12 Hz. Sampling yang terjadi pada gambar 2.23 hanya terjadi sedikit kehilangan informasi



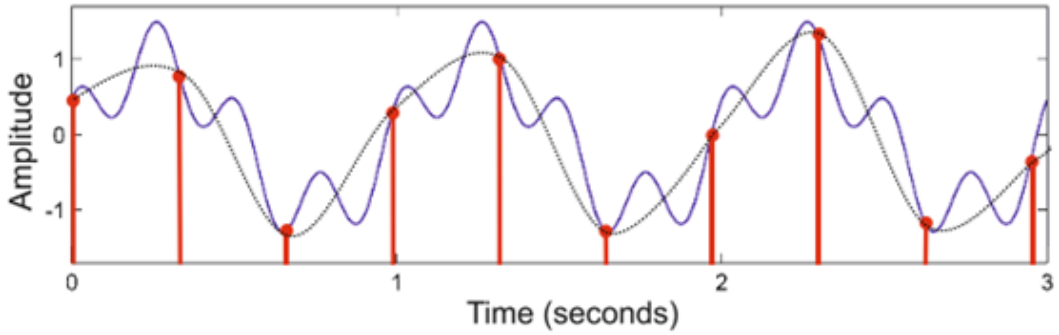
Gambar 2.23 Efek dari aliasing ketika sampling sinyal
Sumber: (Müller, 2015)

Pada gambar 2.24 dengan frekuensi *sampling* 6 Hz terlihat dua puncak dan beberapa pangkal pada sinyal tidak termasuk pada sampel menandakan kehilangan informasi yang cukup signifikan.



Gambar 2.24 Efek dari aliasing ketika sampling sinyal
Sumber: (Müller, 2015)

Hasil *sampling* menjadi 3 Hz pada gambar gambar 2.25 menyebabkan sinyal kehilangan lebih banyak lagi informasi dan kompleksitas sinyalnya dibanding *sampling* sebelumnya.



Gambar 2.25 Efek dari aliasing ketika sampling sinyal
Sumber: (Müller, 2015)

b. *Pre emphasis*

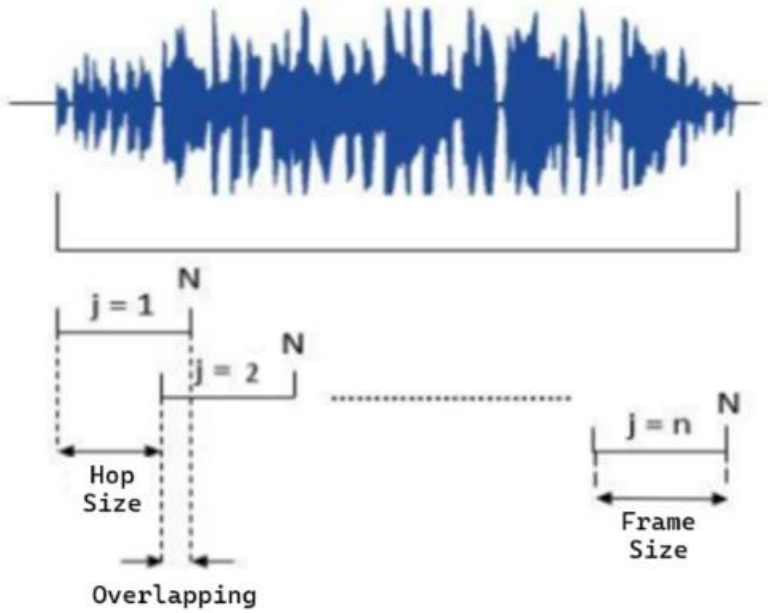
Pre emphasis adalah proses filtering dengan tujuan untuk mendapatkan spektral yang lebih halus dari frekuensi sinyal suara dan mengurangi noise pada saat penangkapan suara. Filter *pre emphasis* didasarkan pada hubungan input/output pada domain waktu dengan persamaan 2.11.

$$y(n) = x(n) - ax(n-1) \qquad (2.11)$$

Dimana a adalah konstanta filter *pre emphasis* dengan rentang $0.9 < a < 1.0$.

c. *Frame Blocking*

Sinyal suara di segmentasikan atau dibagi menjadi beberapa *frame* yang mengandung *sample* tertentu lalu dibuat tumpang tindih (*Overlapping*) satu sama lain dengan jarak yang ditentukan bernama *hop size* (H). Tujuan melakukan *overlapping* untuk meminimalisir sinyal yang terganggu atau kehilangan fitur karena proses *windowing*. Proses ini akan terus berlanjut hingga semua sinyal telah masuk kedalam *frame*. Dalam proses ini, umumnya setiap *frame* memiliki ukuran sekitar 10-30 miliseken karena respon ransangan pendengaran manusia terpendek yang ditemukan ada pada waktu minimal 10 miliseken (Leshowitz, 1970).



Gambar 2.26 *Frame Blocking* pada MFCC
Sumber: (Anantha dkk., 2018)

d. *Windowing*

Windowing adalah tahap dimana setiap *frame* akan diberikan bobot dengan menggunakan fungsi *window*. Proses ini bertujuan untuk mengurangi sinyal yang terputus putus atau memiliki gap pada awal dan akhir atau *endpoint* setiap *frame*,

karena akan menimbulkan noise pada power spectrum. *Window* memiliki syarat $w \leq n \leq N$, dimana N merupakan jumlah sinyal suara pada setiap *frame*, hasil dari proses *windowing* dipresentasikan sebagai:

$$y_w(n) = x(n) \cdot w(n), 0 \leq n \leq N \quad (2.12)$$

$y_w(n)$ merupakan hasil konvolusi antar input sinyal suara didalam *frame* dengan fungsi *window*, $x(n)$ representasi sinyal yang akan dikonvolusi oleh fungsi *window*, dimana $w(n)$ adalah fungsi *window* menggunakan *Hann window* dengan persamaan asal 2.13

$$w(n) = a_0 - (1 - a_0).\cos\left(\frac{2\pi n}{N}\right), 0 \leq n \leq N \quad (2.13)$$

Hann window memiliki aturan dasar bahwa $a_0 = 0.5$, sehingga persamaan dalam bentuk 2.13 menjadi persamaan 2.14

$$w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right), 0 \leq n \leq N \quad (2.14)$$

Atau ketika $a_0 = 25/46 \approx 0.54$ yang dinamakan dengan *Hamming window* menjadi persamaan 2.15

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), 0 \leq n \leq N \quad (2.15)$$

e. *Discrete Fourier Transform (DFT)*

Tahap selanjutnya melakukan ekstraksi informasi spektral untuk mendapatkan informasi daya/energi terhadap setiap frekuensi. Hasil dari tahap ini akan mendapatkan spektrum atau *Spectrum* / *Periodogram*. Input untuk melakukan DFT adalah frekuensi dari n_0 hingga frekuensi ke- N dalam bentuk bilangan real dan memiliki output untuk setiap n berbentuk bilangan komplex X_n dalam bentuk seperti $c = a + bi$ dimana i adalah bilangan imajiner, bilangan kompleks ini

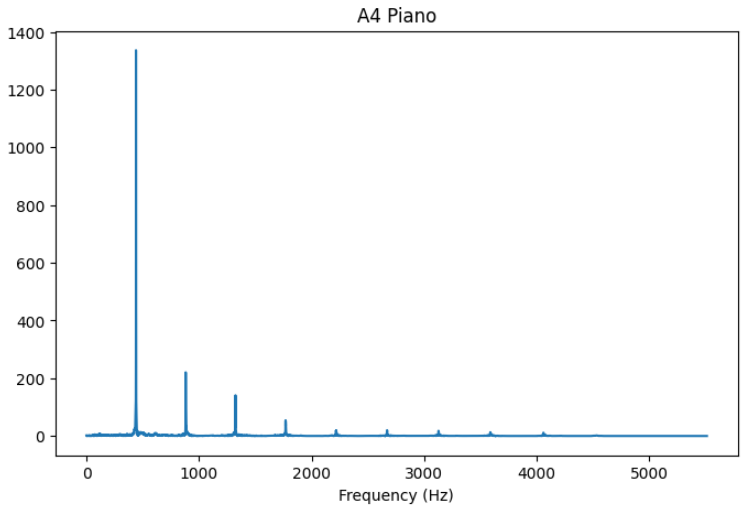
merepresentasikan ukuran (magnitude) dan fase dari komponen frekuensi tersebut terhadap sumber audio.

DFT memiliki formula dalam bentuk seperti pada persamaan 2.16

$$X_{\left(\frac{k}{N}\right)} = \sum_{n=0}^{n=N-1} x(n) \cdot e^{-\frac{i2\pi kn}{N}}, k = [0, M-1] = [0, N-1] \qquad (2.16)$$

$X_{(k/N)}$ adalah bilangan kompleks, M adalah himpunan semua frekuensi pada audio, N adalah himpunan dari total poin waktu audio, n adalah sampel $x(n)$ pada waktu ke-n, dan k adalah subset atau bagian dari frekuensi himpunan K, dimana $K = N$ agar membuat formula teori matrix *fourier transform* tersusun dan membuat algoritma efisien dalam menghitung transformasi.

Contoh hasil plot spektrum audio piano kunci A4 pada gambar 2.27 menampilkan power dari setiap frekuensi yang terdeteksi pada audio, namun informasi yang tidak terlihat adalah informasi mengenai waktu kapan frekuensi yang dimaksud ada, solusinya dengan menggunakan *Short Time Fourier Transform* (STFT).



Gambar 2.27 Spektrum piano kunci A4

Perhitungan dalam STFT hampir serupa dengan bagaimana DFT dilakukan, perbedaan ada pada informasi tambahan pada x sampel seperti yang terlihat pada formula STFT dibawah:

$$S\left(j,\frac{k}{N}\right)=\sum_{n=0}^{n=N-1} x(n+jH)\cdot w(n)\cdot e^{-\frac{i2\pi kn}{N}}, k=[0,M-1]=[0,N-1] (2.17)$$

$S\left(j,\frac{k}{N}\right)$ adalah bilangan komplek pada *frame* ke- j dan frekuensi ke- k, kemudian perjumlahan deret $x(n+jH)$ merupakan sampel ke-n pada frame ke- j dengan pergeseran frame sebesar *hop size* H dengan bobot fungsi windowing $w(n)$.

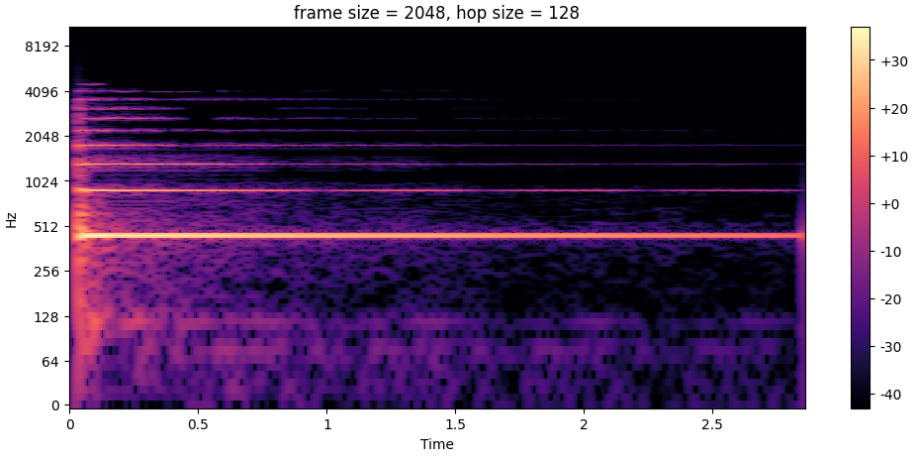
Lalu perbedaan selanjutnya ada pada output yang dihasilkan. DFT memberikan hasil spektrum vektor dari bilangan kompleks dari setiap komponen frekuensi audio, dimensi yang dihasilkan hanya satu, yaitu semua sampel yang dikalkulasi menjadi bilangan kompleks. STFT menghasilkan spektral dalam bentuk dua dimensi sehingga dimensi yang dihasilkan lebih kompleks, yaitu yang pertama mengandung banyaknya frame yang menentukan kualitas dalam sumbu frekuensi, yang kedua frame yang memiliki banyaknya bilangan kompleks sebanyak sampel dalam frame yang menentukan kualitas dalam sumbu waktu.

Perhitungan DFT jika memiliki data dengan jumlah yang besar akan memakan waktu yang sangat lama sekitar 10^N dimana N adalah kelipatan sepuluh dari total data. Dengan memanfaatkan sifat matriks yatu melakukan faktorisasi matriks DFT membagi transformasi menjadi dua bagian dengan ukuran $n/2$ sehingga n harus dalam kelipatan 2, diilustrasikan persamaan 2.18 dibawah ini, komputasi yang diperlukan lebih cepat dengan total sekitar $N\log N$ kali melakukan

perhitungan. Algoritma perhitungan ini dinamakan *Cooley-Turkey FFT* (*Fast Fourier Transform*).

$$DFT_N \cdot \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{pmatrix} = \begin{pmatrix} id_M & \Delta_M \\ id_M & -\Delta_M \end{pmatrix} \begin{pmatrix} DFT_M & 0 \\ 0 & DFT_M \end{pmatrix} \begin{pmatrix} x(0) \\ x(2) \\ \vdots \\ x(N-2) \\ x(1) \\ x(3) \\ \vdots \\ x(N-1) \end{pmatrix} \quad (2.18)$$

Contoh bentuk hasil bilangan komplex DFT dalam bentuk grafik spektogram nada piano A4 terlihat pada gambar 2.28 dengan ukuran frame sebesar 2048 dan ukuran hop sebesar 128, warna yang semakin terang menandakan energi dalam satuan dB yang terdeteksi pada waktu dan frekuensi pada daerah tersebut.



Gambar 2.28 Spektogram piano nada A4

f. *Mel Scalling/Mell-Frequency Wrapping*

Secara garis besar satuan mel merupakan satuan pitch yang ditetapkan dalam mencerminkan persepsi terhadap bagaimana filter pendengaran manusia bekerja, diilustrasikan pada gambar 2.8 mengenai *equal loudness contours*.

Semakin rendah frekuensi, filter akan semakin menyempit, semakin besar frekuensi maka filter semakin renggang. Pendengaran manusia sensitif pada daerah frekuensi rendah terutama dibawah 1000 Hz, namun sulit membedakannya pada frekuensi tinggi.

Persamaan untuk mencari titik - titik skala mel terhadap frekuensi dalam Hz digambarkan pada persamaan 2.19

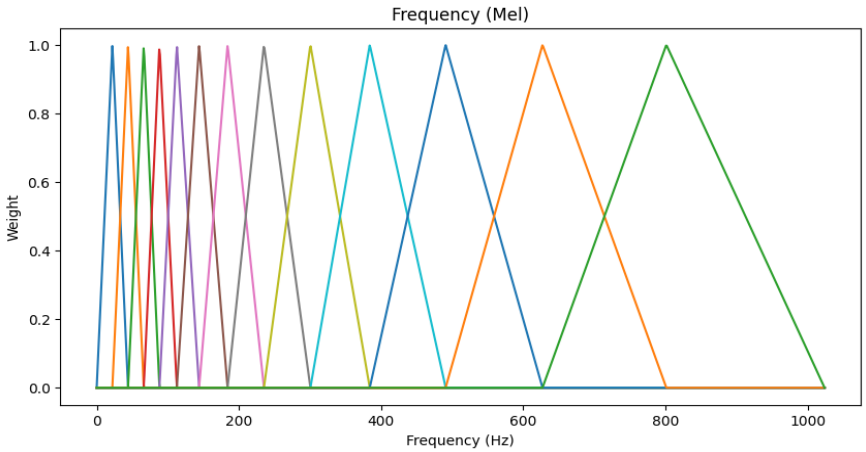
$$F_{mel} = 2595 \cdot [log]_{10}\left(1 + \frac{F_{Hz}}{500}\right) \quad (2.19)$$

Dimana F_{Hz} merupakan frekuensi dalam satuan Hz, dan F_{mel} dalam satuan mel. Kemudian mencari frekuensi sebagai batasan hubungan frekuensi terhadap area filter mel dimana m adalah mel menggunakan persamaan 2.20

$$F_{Hz} = 700\left(10^{\frac{m}{2595}} - 1\right) \quad (2.20)$$

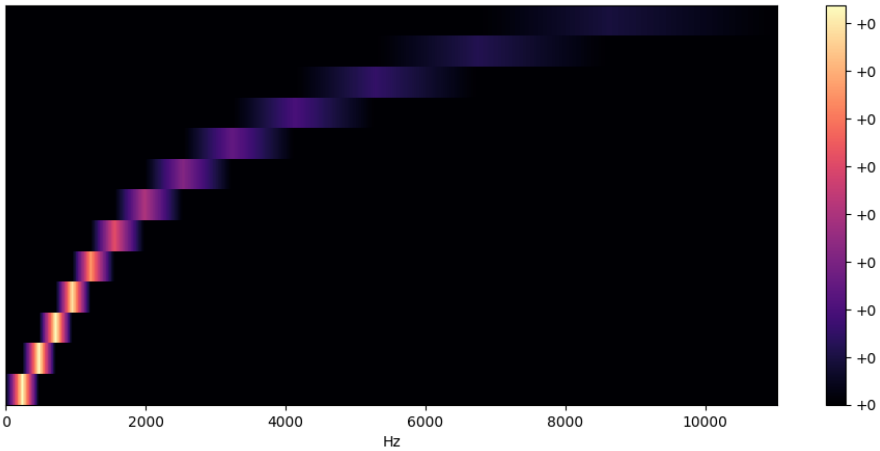
Filter Bank akan digunakan untuk merepresentasikan seberapa besar energi yang ada pada suatu frekuensi. Filter bank melakukan pendekatan spektrum frekuensi dalam satuan mel yang memiliki fungsi kinerja layaknya filter pada telinga manusia.

Pada gambar 2.29 skema mel filter bank memiliki 13 mel filter, masing–masing mel filter melingkupi grup pitch tersendiri, misalnya ketika mendengar suara pada frekuensi 600 Hz, akan dirasa lebih dominan merujuk pada karakter pitch ke-12 dengan sedikit karakteristik pitch ke-11.



Gambar 2.29 Mel-Filter Bank

Mel filter bank bila digambarkan dalam bentuk spektogram akan terlihat seperti pada gambar 2.27 yang memiliki jumlah filter mel serupa sebesar 13.

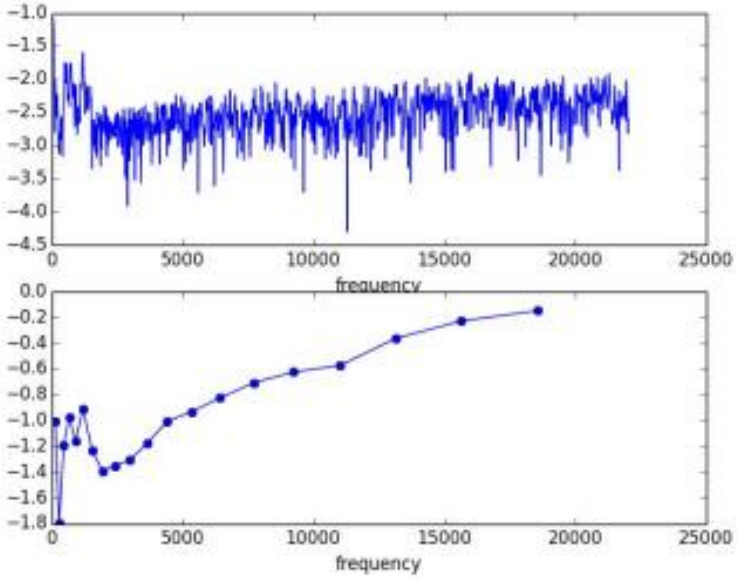


Gambar 2.30 Spektogram Mel Filter Bank

Setelah mel filter bank, himpunan log spektrum frekuensi sampel audio akan dikelompokkan kedalam filter triangular pada proses sekarang yang dinamakan *Mel-frequency wrapping* dengan persamaan 2.21 dimana $i = 1,2,3, \ldots, M$ (M merupakan nilai maksimal filter triangular) dan $H_i(k)$ adalah nilai dari filter triangular ke $-i$ untuk frekuensi audio dari k.

$$X_i = \log_{10}\left(\sum_{k=0}^{N-1}|X(k)|H_i(k)\right) \qquad 2.21$$

Gambar 2.28 dibawah membandingkan amplitudo original sinyal suara dengan sinyal suara yang telah melewati filter mel.



Gambar 2.31 Spektrum amplitudo original sinyal dan Mel bank filter
Sumber: (Marlina dkk., 2018)

g. *Discrete Cosine Transform* (DCT)

Discrete Cosine Transform mengubah bentuk log spektrum frekuensi, yang akan menghasilkan output nilai real *mel frequency cepstrum coefficient* (MFCC) (Sahidullah & Saha, 2012) dengan menyesuaikan log spektrum frekuensi terhadap kosinus dengan frekuensi berbeda–beda. Transformasi kosinus tersebut memiliki persamaan 2.22

$$C_k = \sum_{n=0}^{N-1} X_n \cos\left(\frac{\pi k(2n+1)}{2N}\right) \qquad (2.22)$$

Dimana C_k adalah koefisien MFCC. X_n adalah kekuatan spektrum frekuensi Mel, dimana $k = 1,2,3, \ldots . , N - 1$, N adalah periode.

2.1.7 *Deep Neural Network* (DNN)

Deep Neural Network merupakan salah satu kelas jaringan saraf tiruan (*artificial neural network*). *Artificial Neural Network* atau dalam bahasa indonesia menjadi Jaringan Saraf Tiruan merupakan algoritma komputasi yang terinspirasi dari bagaimana kinerja otak bekerja dalam pengambilan keputusan (Zhang dkk., 2022). Struktur jaringan saraf terdiri dari gabungan *neuron* sebagai unit komputasi yang berhubungan antar lapisannya. Seperti cara kerja jaringan saraf pada otak, jaringan saraf menerima sinyal yaitu data dari jaringan saraf lapisan input lalu memprosesnya menjadi sinyal baru tergantung fungsi aktivasinya yang kemudian diteruskan ke *neuron* lain seperti sinaps pada *neuron* otak makhluk hidup hingga ke lapisan output.

Struktur jaringan saraf tiruan umunya terdiri dari tiga lapisan utama, diantaranya:

a. Lapisan input atau *Input Layer*, yaitu lapisan yang menerima masukan data dari luar baik data mentah atau data yang sudah diolah seperti data extraksi fitur audio.
b. Lapisan tersembunyi atau *hidden layer,* lapisan yang memroses masukan data dan mempelajari fitur–fiturnya. Lapisan ini dapat mengandung lebih dari satu lapisan tersembunyi dan memiliki banyak saraf tiap lapisannya, jaringan syaraf seperti ini dinamakan *Deep Neural Network* (DNN).

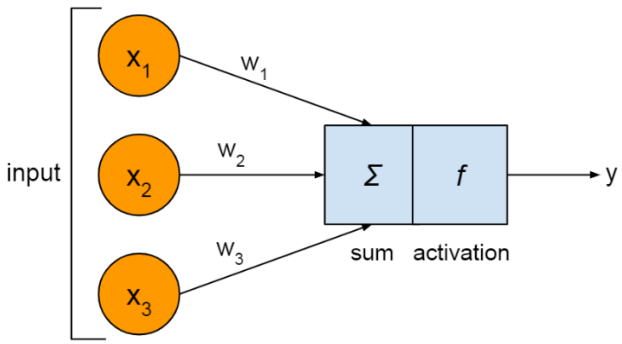
c. Lapisan output atau *Output Layer*, bagian lapisan yang bertanggung jawab dalam memprediksi masukan data.

Secara singkat, alur dalam melatih model jaringan saraf dapat dijabarkan menjadi:

a. Menetapkan Model Jaringan Saraf, dengan memanfaatkan framework dalam membuat saraf pada setiap layer seperti keras, atau MXNet,
b. Inisialisasi model jaringan saraf, mendefinisikan bobot dan bias secara acak,
c. *Forward Propagation*, mulai memberikan data input ke jaringan saraf untuk mendaptkan output dalam bentuk prediksi atau regresi tergantung masalah yang dihadapi,
d. Menghitung error / loss, setelah mendapatkan output dibandingkan dengan label aslinya menggunakan fungsi error,
e. *Backward Propagation*, menghitung gradien error terhadap setiap error dan bias,
f. Update beban dan bias, menyesuaikan nilai beban dan bias guna mengurangi error pada prediksi dengan menggunakan algoritma optimasi,
g. Melakukan pengulangan dalam beberapa iterasi (epoch) hingga jaringan saraf mencapai titik dimana menggapai error yang paling minimal.

Pada struktur jaringan saarf memiliki beban pada setiap hubungan jaringan saraf antar, beban ini merupakan parameter yang disesuaikan ketika tahap latih untuk meminimalisir error ketika prediksi data. Di dalam setiap aktivasi jaringan saraf memiliki pilihan untuk menambahkan bias sebagai parameter tambahan untuk menggeser hasil aktivasi.

Cara kerja jaringan saraf tiruan dapat digambarkan seperti pada gambar 2.32 dimulai dari data input ke salah satu saraf yaitu x_1, x_2 dan x_3 kemudian diteruskan dengan beban w_1, w_2 dan w_3 ke satu saraf kemudian dihitung layaknya jaringan saraf otak diilustrasikan dengan persamaan 2.23, lalu diteruskan lagi sebagai aktivasi jaringan saraf menjadi keluaran $y = f(h)$.



Gambar 2.32 Skema satu Jaringan Saraf
Sumber: (Aggarwal, 2023)

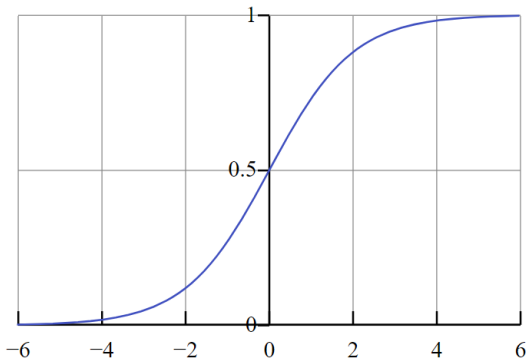
$$h = \sum_i x_i w_i = x_1 w_1 + x_2 w_2 + \cdots + x_i w_i \qquad (2.23)$$

Fungsi aktivasi pada jaringan syaraf berperan penting untuk jaringan syaraf agar bisa belajar pola dari masukan yang diberikan. Ada berbagai macam fungsi aktivasi, yang akan disebutkan disini adalah fungsi aktivasi sigmoid, *Rectified Linear Activation* (ReLU), Softmax, dan *hyperbolic tangent* (Tanh).

Fungsi aktivasi sigmoid memiliki persamaan 2.24

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \qquad (2.24)$$

Aktivasi sigmoid melimitasi output diantara 0 dan 1 dengan karakteristik grafik berbentuk S, sehingga nilai aktivasi berada pada domain bilangan real cocok dalam klasifikasi dua variabel seperti “ya” dan “tidak”.

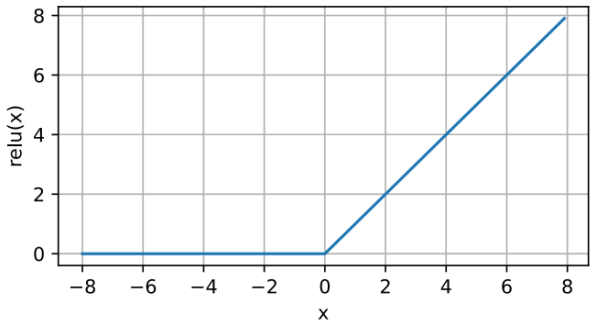


Gambar 2.33 Grafik Kurva Aktivasi sigmoid
Sumber: (Zhang dkk., 2022)

Fungsi Aktivasi ReLU merupakan fungsi aktivasi yang sangat popular digunakan karena kemudahan implementasi dan memiliki performa yang bagus dalam masalah prediktif, didefinisikan dalam persamaan 2.25

$$ReLU(x) = \max(x, 0) \qquad (2.25)$$

Fungsi ReLU hanya mengembalikan nilai elemen positif dan nol jika menerima nilai elemen negatif sesuai dengan pengaturan fungsi persamaan diatas. Ilustrai dalam grafik ada pada gambar 2.34.

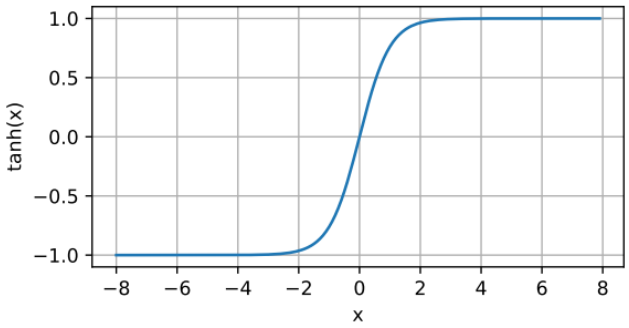


Gambar 2.34 Grafik aktivasi ReLU
Sumber: (Zhang dkk., 2022)

Fungsi Aktivasi *hyperbolic tangent* memiliki bentuk seperti fungsi sigmoid yang memeras input dn mengubahnya menjadi elemen dengan interval kecil, untuk fungsi tanh antara -1 dan 1, didefinisikan dalam persamaan 2.26

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{2.26}$$

Perbedaan dengan sigmoid terlihat ketika lekukan grafik mulai mendekati 0, nilainya berubah lebih tajam dibanding dengan sigmoid.



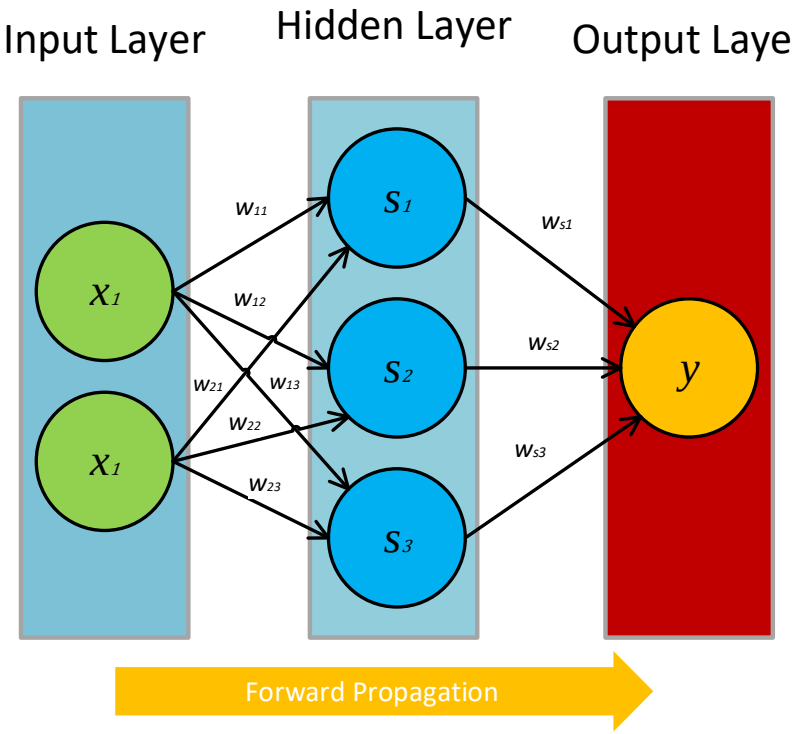
Gambar 2.35 Grafik Aktivasi Tanh
Sumber: (Zhang dkk., 2022)

Fungsi Softmax merupakan fungsi yang dipakai pada jaringan saraf pada layer output. Guna mengoptimalkan parameter dalam menghasilkan nilai probabilistik yang dapat memaksimalkan kemungkinan menghasilkan prediksi. Input ke dalam fungsi softmax adalah hasil dari fungsi aktivasi dari jaringan saraf sebelumnya, outputnya adalah matriks probabilitas prediksi berdimensi seperti *one-hot encoding* yaitu vektor yang mewakili setiap kelas pada dataset. Fungsi softmax dapat di expresikan dalam persamaan 2.27.

$$Softmax(X_{ij}) = \frac{e^{X_{ij}}}{\sum_{k=1}^{K} e^{X_{ik}}} \tag{2.27}$$

Proses dimana penyerahan hasil perhitungan pada saraf secara bertahap dari lapisan input hingga ke lapisan output dinamakan *forward propagation*. Asumsi

contoh input x dan bilamana bias tidak termasuk didalam *hidden layer*, dapat diilustrasikan seperti pada gambar 2.33.



Gambar 2.36 Skema Jaringan Saraf Forward Propagation
Sumber: (Abiodun dkk., 2019)

Semua data input akan dibobotkan dengan bobot w sebelum dilanjutkan ke jaringan saraf pada hidden layer, sehingga notasi lanjutan inputnya menjadi persamaan 2.28.

$$S = \sum_{i=1}^{n} w_i x_i \qquad (2.28)$$

Dimana S adalah data pada posisi hidden layer S, X adalah input data, $W^{(1)}$ adalah bobot dan i adalah jumlah terusan ke jaringan saraf selanjutnya. Dalam notasi matriks untuk jaringan saraf gambar 2.33 menjadi persamaan matriks 2.29

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.29)$$

Hasil perhitungan bobot data terusan kemudian diteruskan lagi ke dalam fungsi aktivasi di setiap jaringan saraf pada hidden layer, sehingga bentuknya menjadi $f(s_\mathrm{i})$ dimana i adalah jaringan saraf ke-i. Hasil aktivasi diteruskan ke jaringan saraf output, sehingga akan berbentuk menjadi persamaan 2.30

$$[y] = [f(s_1) \quad f(s_2) \quad f(s_3)] \begin{bmatrix} w_{\mathrm{s}1} \\ w_{\mathrm{s}2} \\ w_{\mathrm{s}3} \end{bmatrix} \quad (2.30)$$

Langkah terakhir menerapkan fungsi output, pada kasus ini fungsi softmax, maka output model menjadi persamaan 2.31

$$\boldsymbol{O} = f(y) \quad (2.31)$$

Lalu menghitung error atau loss dengan ekpresi matematika persamaan 2.32 menggunakan fungsi error dimana $\boldsymbol{O}$ adalah hasil prediksi, dan *P* prediksi yang diharapkan atau dapat dikatakan sebagai label sebenarnya.

$$E = f_\mathrm{E}(\boldsymbol{O}, P) \quad (2.32)$$

Beberapa jenis fungsi error seperti fungsi MSE (*Mean Square Error*) dengan pesamaan 2.33 merupakan fungsi error yang cocok untuk model regresi.

$$E = \frac{1}{N}\sum_{i=0}^{N}(\boldsymbol{O}_i - P_i)^2 \quad (2.33)$$

Fungsi *Cross Entropy* dengan persamaan 2.34 merupakan fungsi yang cocok dalam melakukan prediksi probabilitas dalam klasifikasi, adapun modifikasi *Cross Entropy* ketika memasuki masalah klasifikasi kelas ganda menjadi persamaan 2.35.

$$E = -\sum_{i=0}^{N} P_i \log(\boldsymbol{O}_i) \quad (2.34)$$

$$E = -\frac{1}{N}\sum_{i=0}^{N} P_i \log(\boldsymbol{O}_i) + (1 - P_i)\log(1 - \boldsymbol{O}_i) \qquad (2.35)$$

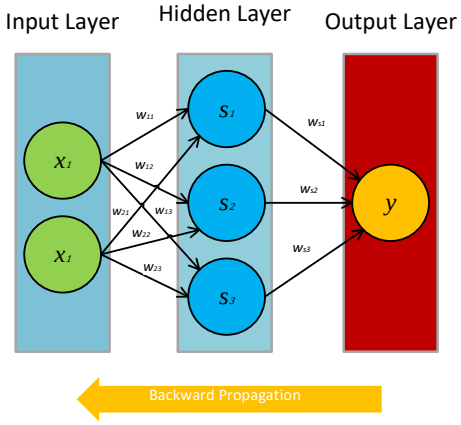
Fungsi error ketiga yaitu fungsi MAPE (*Mean Absolut Percentage Error*) biasa digunakan untuk mengukur performa jaringan saraf ketika melakukan prediksi dengan persamaan 2.36

$$MAPE = \frac{100\%}{N}\sum_{i=0}^{N} \frac{|\boldsymbol{O}_i - P_i|}{P_i} \qquad (2.36)$$

Backpropagation merupakan metode dalam menghitung parameter gradien error dari jaringan syaraf, yaitu dengan alur sebaliknya dari *forward propagation* dari output ke input layer, guna memperbarui parameter bobot dan bias jika digunakan pada jaringan syaraf. Umumnya dapat di ekspresikan menjadi persamaan 2.37

$$\frac{\partial E}{\partial W^{(n)}} \qquad (2.37)$$

Dimana ∂E adalah turunan fungsi error, dan $\partial W^{(n)}$ merupakan bobot ke-n yang akan diubah. Alur melakukan *backward propagation* seperti pada gambar 2.37.



Gambar 2.37 Backward Propagation
Sumber: (Abiodun dkk., 2019)

Memperbarui bobot pada jaringan saraf tidak bisa dilakukan secara sembarang karena berpotensi meningkatkan jarak error terhadap prediksinya, maka digunakan algoritma optimasi untuk membantu memperbarui bobot dan bias ke arah yang dapat mengurangi tingkat error pada prediksi. Beberapa algoritma optimasi yang sering dipakai antara lain:

a. *Gradient Descent*, secara berulang memperbarui bobot dan bias, memiliki beberapa tipe seperti:
    1) *Batch Gradient Descent*, menggunakan seluruh dataset dalam menghitung gradien dan memperbarui parameternya, akan bekerja dengan lamban untuk ukuran dataset yang besar,
    2) *Stochastic gradient descent* (SGD), menggunakan dataset secara acak dalam menghitung gradien dan memperbarui parameter, karena secara acak dapat membantu kecepatan perhitungan dan update, namun menimbulkan fruktuasi pada fungsi error,
    3) *Mini-Batch Gradient Descent*, gabungan dari kedua jenis *gradient descent* dengan menggunakan sebagian dataset latih secara acak dalam menghitung gradien dan memperbarui parameter.
b. *Momentum*, menggunakan nilai dari gradien sebelumnya sebagai pertimbangan dalam menghitung gradien selanjutnya, menyebabkan perhitungan yang lebih cepat dan memperlancar pembaruan parameter,
c. RMSprop (*Root mean square propagation*), menggunakan gradien kuadrat terbaru dalam mengadaptasi tingkat belajar jaringan saraf untuk pembaruan

setiap parameter, membantu mengurangi isu gradien yang terjal dan menormalisasikan tingkat belajar jaringan saraf.

d. *Adaptive Moment Estimator* (Adam) bekerja dengan cara menggabungkan kelebihan pada momentum dan RMSprop.

Selama model jaringan saraf melakukan sesi latih, ada kemungkinan model bukan belajar pola dari data untuk klasifikasi kelasnya tetapi mengingat data dalam klasifikasi ke kelas yang dituju, masalah ini dinamakan *overfitting*, atau bahkan model selesai dalam tahap belajar namun . Terdapat beberapa istilah terkait topik ini, diantaranya *training error* dan *generalization error*. *Training error* terjadi ketika model melakukan perhitungan pada *dataset* latih, sedangkan *generalization error* terjadi ketika model diberi dataset dengan dasar yang sama secara terus menerus. Beberapa aspek yang mempengaruhi *generalization error* antara lain:

a. Jumlah parameter yang bisa di tuning terlalu banyak
b. Nilai parameter memiliki rentang nilai yang besar
c. Jumlah pelatihan model yang dilakukan terlalu banyak ketika dataset sedikit.

Pada saat membandingkan *error training* dan *error* validasi, hal yang perlu diperhatikan adalah ketika nilai *error* keduanya besar dan memiliki selisih yang kecil, lalu model tidak bisa lagi mengurangi *error training* kemungkinan karena bentuk model yang terlalu simpel atau kurang ekspresif, atau gagal dalam menangkap pola pada data, hal ini dinamakan *underfitting*.

Strategi yang bisa digunakan dalam menghadapi *overfitting* maupun *underfitting* dapat dimulai setelah rekonstruksi model, menambah atau mengurangi

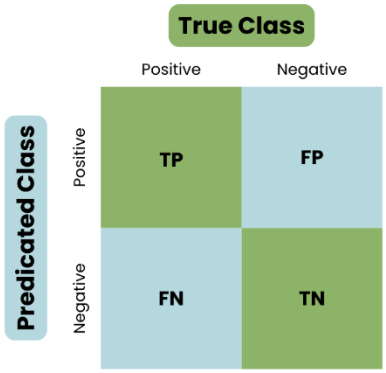
lapisan atau saraf. Menggunakan parameter *regularization* untuk memberi pelanggaan kepada beban yang terlalu besar, bila digunakan pada fungsi *error* MSE maka akan menjadi seperti pada persamaan 2.38 dengan contoh pada fungsi error MSE dimana T adalah layer.

$$E = \frac{1}{N}\sum_{i=0}^{N}(\boldsymbol{O}_i - P_i)^2 + \lambda \sum_{i=1}^{N} w_i^T \qquad (2.38)$$

Menambah atau mengurangi fungsi *dropout* secara acak. Melakukan *early stopping* dengan memonitor *error* validasi jika sudah mulai meningkat atau tidak ada perkembangan. Augmentasi data untuk menambahkan variasi ciri pada dataset, atau bahkan bisa dengan menambah dataset baru.

2.1.8 Confusion Matrix

Confusion matrix merupakan alat evaluasi performa model dan umumnya disajikan dalam bentuk tabel. Hasil penyajiannnya memberikan wawasan mendalam terhadap performa model, error dan kelemahannya. Struktur confusion matrix berbentuk seperti pada gambar 2.38 yang berisi *true positive* (TP), *true negative* (TN), *false positive* (FP) dan *false negative* (FN).



Gambar 2.38 Tabel confusion matrix
Sumber: (Nisha Arya Ahmed, 2023)

Nilai pada *true positive* (TP) menandakan banyaknya prediksi kelas positif pada kelas yang benar, contohnya benar dalam mengidentifikasi spam email pada kategori spam. Nilai pada *false positive* (FP) menandakan banyaknya prediksi kelas negatif pada kelas yang benar, contohnya benar dalam prediksi email biasa pada kategori email biasa. Nilai pada *false positive* (FP) menandakan banyaknya prediksi kelas positif pada kelas yang salah, contohnya positif mengidentifikasi email biasa ke dalam kategori spam. Nilai pada *false negative* (FN) menandakan banyaknya prediksi kelas negatif pada kelas yang salah, contohnya positif mengidentifikasi email spam ke dalam kategori bukan spam. Keempat jenis nilai pada tabel confusion matrix akan berguna untuk mengukur *accuracy, precision, recall,* dan *F1 Score*.

Accuracy atau akurasi yaitu sesuai dengan namanya, mengukur tingkat akurasi keseluruhan klasifikasi dengan menghitung total data diindetifikasikan dengan benar sesuai dengan kelasnya dibagi dengan total prediksi yang dilakukan. Akurasi dapat diekspresikan dengan persamaan 2.39. Akan tetapi hanya dengan mengandalkan akurasi saja tidak cukup sebagai ukuran penilaian bagus tidaknya model terutama jika dataset tidak seimbang, yang artinya penyebaran setiap data memiliki nilai yang berbeda, contohnya ketika hasil klasifikasi ada *false positive* (FP) dan *false negative* (FN), akurasi tidak menilai kedua aspek ini maka diperlukan ukuran lain yaitu *precision* dan *recall*.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{TP + TN + FP + FN} \qquad (2.39)$$

Precision atau presisi menilai tingkat kebenaran model dalam melakukan klasifikasi terhadap satu kelas. Presisi dapat diekspresikan dengan persamaan 2.40. Idealnya presisi bernilai satu untuk model dapat dinilai baik dalam melakukan klasifikasi, yang artinya tidak ada *false positive* pada klasifikasi kelas tersebut sehingga model persisi dalam melakukan klasifikasi.

$$Precision = \frac{\mathrm{TP}}{TP + FP} \tag{2.40}$$

Recall atau secara harfiah dapat diartikan mengingat kembali, menunjukkan tingkat kebenaran model dalam klasifikasi satu kelas terhadap kelas itu sendiri. *Recall* dapat diekspresikan dengan persamaan 2.41. Idealnya *recall* bernilai tinggi (satu) untuk model agar dapat dinilai baik dalam melaukan klasifikasi, yang artinya model sangat mampu dalam mengingat kembali kelas yang dimaksud.

$$Recall = \frac{TP}{TP + FN} \tag{2.41}$$

Berdasarkan presisi dan *recall* model klasifikasi akan sangat baik dalam melakukan tugasnya ketika *false negative* dan *false positive* semakin kecil. Karena itu diperlukan pengukuran yang mempertimbangkan presisi dan *recall* yaitu *F1-score*. *F1-score* dapat diekspresikan dengan persamaan 2.42. nilai *f1-score* akan bernilai satu ketika presisi dan *recall* sama sama bernilai satu.

$$F\ Score = \frac{2 \times recall \times precision}{recall + precision} \tag{2.42}$$

2.2 State of the Art

Tabel 2.2 dibawah ini merupakan beberapa penelitian terkait dengan pengolahan data audio untuk mendapatkan cirinya, dan bidang klasifikasi dalam data audio.

Tabel 2.2 Penelitian terkait dan Keterbauan Penelitian

No.	Penulis dan Tahun Penelitian	Judul	Permasalahan	Metode/Solusi	Hasil Penelitian
1.	(Adhinata dkk., 2021)	Pengenalan Jenis Kelamin Manusia Berbasis Suara menggunakan MFCC dan GMM	Seringkali data biometrik suara manusia dari laki – laki menyerupai perempuan, dan juga sebaliknya sehingga sukar dibedakan	- Ekstraksi MFCC - Model klasifikasi Gaussian Mixture Model (GMM)	- Dataset yang digunakan diambil dari dataset publik googleAudioSet yang terdiri dari 5 suara laki – laki dan 5 suara perempuan - Hasil dari klasifikasi data biometrik suara manusia menghasilkan akurasi 81,18% dengan variasi mixture GMM sebesar 16
2.	(Putra, Pradana, dkk., 2021)	Klasifikasi Hukum Bacaan Mad	Klasifikasi hukum bacaan mad pada Surah Maryam ayat	- Fitur Estraksi MFCC	Model memiliki akurasi tertinggi saat diuji per kata sebesar 80%, sedangkat saat diuji per kalimat sebesar 44%,

		Menggunakan Mel Frequency Cepstal Ceofficient (MFCC) dan Hidden Makrov Model (HMM)	1-15 oleh sembilan qori yang dipilih oleh Universitas Darussalam	- Model Klasifikasi Hidden Makrov Model (HMM)	penulis klaim hal ini disebabkan karena tidak adanya nada pada data train, sedangkat data test memiliki nada.
3.	(Sharif dkk., 2023)	*Analysis of variation in the Number of MFCC Features in Contrast to LSTM in the Classification of English Accent sound*	Klasifiksi aksen bahasa inggris oleh 2172 pengucap dari dataset publik kaggle dari 177 negara dan 214 bahasa ibu yang berebeda dari kelamin pria dan wanita, mengucapkan paragraf yang sama	- Fitur ekstraksi MFCC - Model Klasifikasi Long Short-Term Memory (LSTM)	- Sebanyak 2172 sampel audio lelaki dan wanita dipisah dengan perbandingan 80:20 sebagai data latih dan data uji - Kalimat yang diucapkan pada setiap sampel yaitu "*Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy*

					frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station" - Nilai koefisien mel yang disaranan sekitar 12 hingga 20 - Hasil uji model klasifikasi memperlihatkan rata rata akurasi sekitar 55%-62% di rentang koefisien mel 12-20 - Simpulan ditulis bahwa LSTM kurang sesuai dipakai untuk pengenalan aksen - Disarankan data dibagi menjadi kata per kata guna meningkatkan akurasi
4.	(Triandi dkk., 2021)	Perbandingan Teknik Ekstrak Ciri Suara	Perlunya gambaran dari metode ekstraksi ciri yang	- Estrasi ciri MFCC	- Membandingkan tingkat keakuratan ekstraksi ciri audio dan kecepatan

		Pembicara Antara Metode MFCC dan LPC untuk Pengenalan Suara	efektif untuk identifikasi ciri yang mengarah ke pengenalan suara	- Ekstraksi ciri Linear Predictive Coding (LPC)	waktu ekstraksi antara metode MFCC dengan LPC - Digunakan metode *Principal Component Analysis* (PCA) untuk mereduksi vektor ciri sinyal suara dalam dimensi tinggi sehingga perhitungan tetap dapat dilakukan secara optimal untuk kedua metode yang akan dibandingkan - Dari hasil perbandingan memberikan gambaran bahwa metode MFCC membutuhkan waktu ekstraksi ciri yang lebih singkat, namun memerlukan waktu training yang sedikit lebih lambat dibanding LPC
5.	(Marlina dkk., 2018)	*Makhraj* Recognition of *Hijaiyah* Letter	Pengenalan Makhraj huruf	- Fitur ekstraksi MFCC	- Penelitian ini menunjukkan bahwa setiap huruf hijaiyah dapat dibedakan satu dengan yang lain

		for Children Based on Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machines (SVM)	hijaiyah untuk pendidikan anak	- Model klasifikasi Support Vector Machine (SVM)	sekalipun memiliki pengucapan yang sama, seperti ح (”ha”) dan ه (”Ha”) yang terlihat perbedaannya pada analisis FFT dan mel - Pada klasifikasi fitur *Makhraj Hijaiyah* menggunakan SVM, dari 12 perbandingan fitur, hanya ada 2 perbandingan yang sulit dibedakan dikarenakan jarak target ekstraksi figur yang sangat berdekatan, sehingga dapat disimpulkan pengucapan *Makhraj Hijaiyah* dapat diklasifikasikan
6.	(Fathurrahman dkk., 2018)	Deep Neural Network untuk Pengenalan Ucapan Pada Bahasa Sunda Dialek Tengah	Sedikitnya riset *Automatic Speech Recognition* (ASR) dalam mengenali bahasa sunda	- Ekstraksi fitur MFCC - Model klasifikasi DNN	- Untuk mendapatkan akurasi yang lebih baik pada model klasifikasi DNN dapat menggunkan L2 Weight Regulazitation dengan nilai sekitar $n \times 10^{-3}$ atau $n \times 10^{-4}$

		Timur (Majalengka)			- Jumlah dan ukuran saraf pada *hidden layer* lebih baik tidak terlalu banyak, lebih simpel lebih baik.
7.	(Muda dkk., 2010)	*Voice Recognition Algorithms Using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques*	Perlunya pembuktian bahwa ekstraksi fitur MFCC dapat benar – benar bekerja sebagai teknik untuk ekstraksi fitur	- Ekstraksi fitur MFCC - Pencocokan fitur *Dynamic Time Warping* (DTW)	- DTW berkemampuan untuk mendapatkan penyelarasan optimal pada kedua deret waktu yang salah satunya di perpanjang maupun di perpendek, lalu mencocokkan wilayahnya yang sesuai - Hasil penelitian menyatakan bahwa metode MFCC terbukti bisa digunakan untuk penelitian pengenalan suara setelah melakukan pencocokan fitur menggunakan DTW ketika mencocokkan suara pada referensi template dengan suara yang diterima dari luar.

8.	(Abiodun dkk., 2019)	Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition	Kebutuhan akan penerapan State-of-the-Art *Neural Network* pada PR untuk segera mengatasi masalah kurangnya literatur mengenai isu-isu fokus penelitian dan kemajuan pada area *pattern recognition*.	- Generative Adversarial Learing Deep Neural Network, Sparse Auto-Encoder (GALDNN) - Deep Belief Net (DBN) - Convolutional Neural Network (CNN) - Recurrent Neural Network (RNN) - Probabilistic Neural Network (PNN)	- Review mengenai penerapan ANN untuk pengenalan pola, menjelaskan tentang permasalahan yang masih da seperti klasifikasi objek, lokasi, skala, analisis perilaku neuron di dalam lapisan tersembunyi, dan aturan & pencocokan template. - Banyaknya penerapan evaluasi performa model ANN dengan menggunakan *Mean Absolute Percentage Error* (MAPE), *Mean Absolut Error* (MAE), *Root Mean Squard Error* (RMSE), *Variance of Absolute Percentage Error* (VAPE) - Berbagai jenis model ANN seperti *Generative Adversarial Learing Deep Neural Network*, *Sparse Auto-Encoder* (GALDNN), *Deep Belief Net* (DBN), *Convolutional Neural Network* (CNN),

				- Time-Delay Neural Network (TDNN) - Self-Organizing Map (SOP) - Transformer Model	*Recurrent Neural Network* (RNN), *Probabilistic Neural Network* (PNN), *Time-Delay Neural Network* (TDNN), *Self-Organizing Map* (SOP), dan *Transformer Model* bekerja dengan baik dalam penerapannya pada permasalahan pengenalan pola
9.	(Nanni dkk., 2021)	*An ensemble of convolutional neural networks for audio classification*	Studi secara luas mengenai Klasifikasi dengan (CNN) tentang penerapan pengenalan audio suasana/lingkungan yang sering digunakan sebagai pelatihan noise pada model klasifiasi spesies	- Model klasifikasi CNN yang telah dilatih seperti AlexNet - GoogleNet - VGGNet - ResNet - InceptionV3	- Dataset yang akan dites ada tiga yaitu burung, kucing, beserta dengan dataset *environmental sound classification* (ESC-50) sebagai dataset pengenalan sumber noise suara di suatu lingkungan. - Enam kumpulan data augmentasi diterapkan yaitu *Standard Signal Augmentation* (SGN), *Short Signal Augmentation* (SSA), *Super Signal Augmentation* (SSiA), *Time-Scale Modification* (TSM), *Short*

			hewan langka di alam liar atau identifiasi suara mengganggu di area perkotaan		*Spectrogram Augmentation* (SSpA), dan *Super Spectro Augmentation* (SuSA) - menggunakan gabungan model klasifikasi CNN yang digabungkan mendapatkan 97% akurasi pada dataset burung, 90.51% pada dataset kucing, dan 88.65% pada dataset ESC-50 - Dibandingkan dengan lima CNN yang sudah dilatih, memiliki hasil akurasi yang serupa selain dataset noise yang hanya kehilangan akurasi sebesar 0.1%
10.	(Phan dkk., 2017)	*Audio Scene Classification with Deep Recurrent Neural Networks*	Klasifikasi audio adegan yang efisien menjadi tren baru karena perpindahan teknik klasifikasi	- Model klasifikasi *Recurrent Neural Network* dengan kalibari linear SVM	- Dataset yang diambil dari LITIS-Rouen. - Melakukan 5 klasifikasi berbeda dengan gabungan ekstraksi ciri dengan bantuan *Label Tree*

			konvensional ke ranah lebih modern menggunakan *deep learning*, pendekatan yang efisien menjadi sebuah perlombaan		*embedding* (LTE) menghasilkan 3 set. Set pertama dengan 64 *Gammatone Cepstral Coefficient* (Gam), 60 MFCC, dan set ketiga 20 koefisien log-frequency filter bank. Klasifikasi keempat melatih dengan menggabungkan ketiganya dinamakan RNN-fusion, lalu klasifikasi kelima dengan multi-stream RNN - Masing masing memiliki f1 score yang tinggi dengan urutan yang sama, yaitu 96.6%, 95.8%, 96.2%, 97.3%, 97.7%
11.	(Omran dkk., 2023)	*Automatic Detection of Some Tajweed Rules*	Pengenalan terhadap hukum *qalqalah* yang mengandung lima	- Fitur ekstraksi MFCC - Model klasifikasi CNN	- Dataset mengandung empat qori dengan total data sebanyak 3322 sampel pengejaan hukum *qalqalah* - Fitur ekstraksi dengan MFCC disimpan dalam bentuk gambar

			huruf vokal *Qalqalah*		spektrum dari spektrum atau disebut cepstrum - Hasil akurasi validasi model dengan CNN mengandung empat hidden layer, dua filter Max-pooling memiliki hasil 81,1%
12.	(Putra, Musthafa, dkk., 2021)	Klasifikasi Intonasi Bahasa Jawa Khas Ponogoro Menggunakan Algoritma Multilayer Perception Neural Network	Melakukan klasifikasi intonasi bahasa jawa khas ponogoro yang belum dilakukan sebelumnya pada masa penelitian ini dibuat	- Fitur ekstraksi MFCC - Model klasifikasi DNN	- Melakukan ekstraksi ciri dengan MFCC dengan dataset tuturan bahasa jawa ponogoro dengan sample rate 4410 Hz - Klasifikasi menggunakan DNN dengan fungsi dropout untuk setiap hidden layernya - Akurasi yang didapat sebesar 81.25%
13.	(Alsahafí & Asad, 2024)	*Empirical Study on Mispronunciation Detection for*	Kurangnya perhatian terhadap bidang *speech recognition* pada	- Fitur ekstraksi MFCC - Penggambaran fitur Spektogram	- Melakukan penelitian empiris terhadap deteksi salah eja Al-quran dengan menerapkan tiga jenis bentuk fitur ekstraksi

		Tajweed Rules during Quran Recitation	konsentrasi deteksi salah eja Al-Qur'an	- Ekstraksi pitch dengan kaldi pitch - Model klasifikasi Gradient Boosting (GB), - Support Vector Machines (SVM), - Logistic Regression (LR), - Random Forest (RF), - Artificial Neural Network (ANN), - Convolutional Neural Networks (CNN), - Long Short-Term Memory (LSTM)	- Dataset yang digunakan adalah dataset QDAT yang berisikan 1500 sampel tiga jenis ejaan hukum tajwid yang benar yaitu hukum tajwid mad jaiz munfasil, ghunnah, dan ikhfa. - Dari perbandingan hasil ekstraksi fitur, fitur ekstraksi MFCC lebih unggul jika digunakan pada model klasifikasi *deep learning* - CNN terbukti efektif dalam menangkap fitur pada spektogram pada hukum tajwid ikhfa, selebihnya secara rata-rata GB, SVM, LR, dan RF lebih unggul dibanding *deep learning* - Pada kaldi pitch cukup merata pada model *machine learning* dan *deep learning*

14.	(Alrumiah & Al-Shargabi, 2023)	*A Deep Diacritics-Based Recognition Model for Arabic Speech: Quranic Verses as Case Study*	Sebagian besar model pengenalan ucapan berbasis bahasa Arab membuang diakritik, sehingga perlunya pengenalan suara arab klasik yang dapat diubah menjadi teks arab dengan tanda diakritik	- Ekstraksi fitur MFCC - Model klasifikasi Time Delay Neural Network (TDNN)-Connectionist Temporal Classification (CTC), - Recurrent Neural Network (RNN)-CTC, dan - Transformer Model	- Dataset rekaman audio sebesar 72,735 arab klasik yang dibagi menjadi data latih, data validasi, dan data uji - RNN-CTC lebih uggul dibanding dengan kedua model klasifiksi lain karena memiliki *word error rate* (WER) sebesar 19,43%, sedangkan klasifikasi TDNN-CTC sebesar 45,73%, lalu Transformer model 95,03%

2.3 Matrix State of the Art

Tabel 2.3 merupakan daftar matriks penelitian yang telah dilakukan oleh peneliti sebelumnya yang memiliki topik serupa mengenai klasifiksi terhadap suara yang bersumber dari pita suara manusia ataupun memiliki objek serupa yaitu bagian lantunan bahasa arab pada Al-Qur'an.

Tabel 2.3 Matriks State of the Art

No.	Penulis	Ruang Lingkup				
		Klasifikasi				
		HMM	SVM	DNN	CNN	LSTM
1	*Mad Reading Law Clasification Using Mel Frequency Cepstral Coefficient (MFCC) and Hidden Makrov Model (HMM)* (Putra, Pradana, dkk., 2021)	✓				
2	*Makhraj Recognition of Hijaiyah Letter for Children Based on Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machines (SVM Method)* (Marlina dkk., 2018)		✓			

3	Klasifikasi Intonasi Bahasa Jawa Khas Ponogoro Menggunakan Algoritma Multilayer Perception Neural Network (Putra, Musthafa, dkk., 2021)			✓		
4	*Automatic Detection of Some Tajweed Rules* (Omran dkk., 2023)				✓	
5	*Empirical Study on Mispronunciation Detection for Tajweed Rules during Quran Recitation* (Alsahafí & Asad, 2024)			✓	✓	✓
6	(Penelitian ini) Klasifikasi Sinyal Audio Menggunakan Mel-Frequency Cepstral Coefficients dan Deep Neural Network untuk Dataset Hukum Tajwid Nun Sukun atau Tanwin			✓		