



# Audio Coding & Compression

## Group 3

Edgrant H. S. (2206025016)

Giovan C. S. (2206816084)

Nicholas S. (2206059396)



**01**

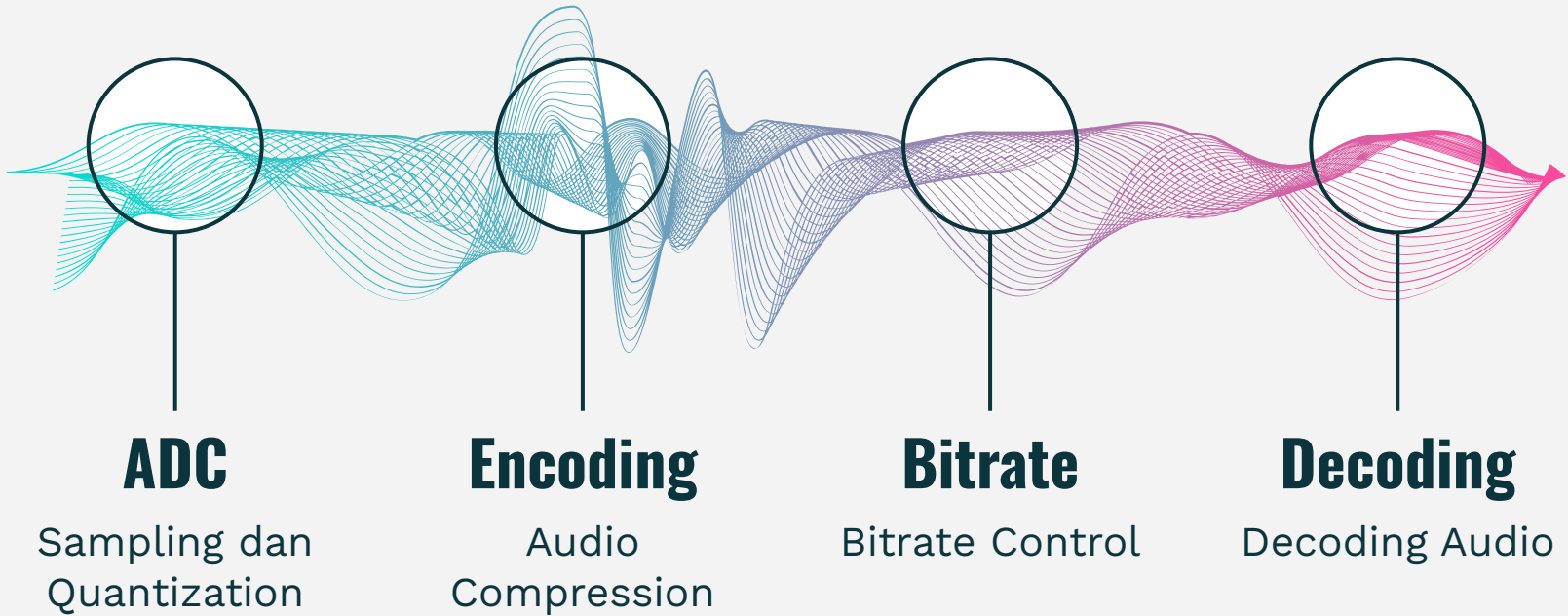
# **Introduction to Audio Coding**

# Apa Itu Audio Coding

Audio coding adalah proses pengubahan data dalam bentuk audio menjadi bentuk lain yang lebih efisien dan dapat disimpan dalam media penyimpanan yang lebih kecil. Audio coding juga dikenal sebagai audio compression. Proses ini sangat penting dalam pengolahan sinyal audio karena memungkinkan untuk menyimpan lebih banyak data audio dalam media penyimpanan yang sama dan juga mentransmisikan data audio melalui jaringan dengan kecepatan yang lebih tinggi.

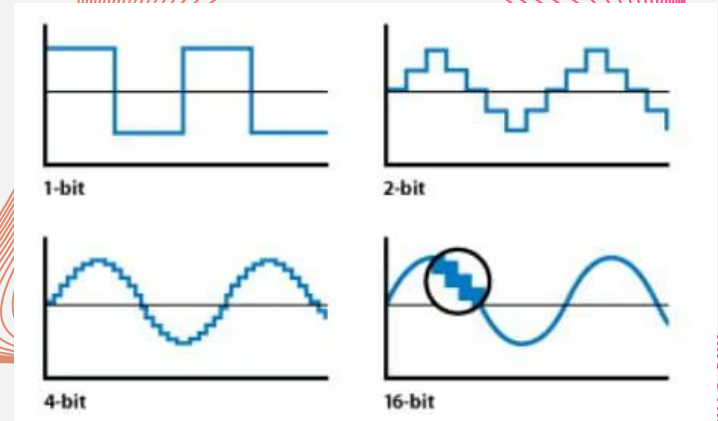


# Cara Kerja Audio Coding



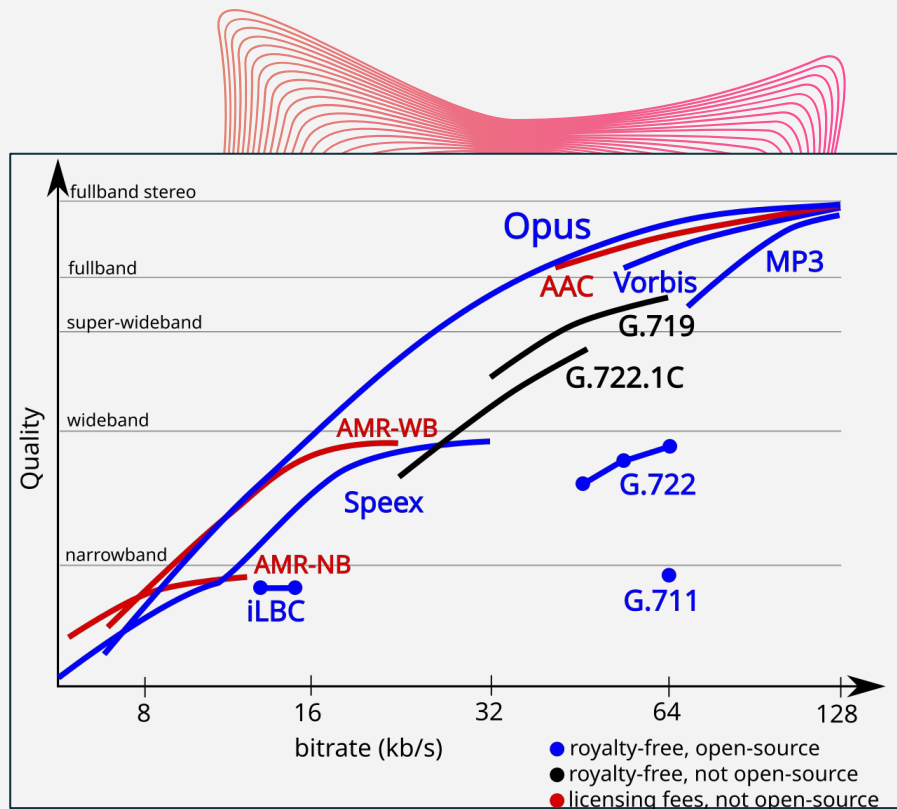
# ADC

Signal suara akan diubah menjadi sinyal digital dengan sampling rate tertentu dan dilakukan quantization. Dalam kata lain, signal analog suara akan diambil cuplikannya pada interval tertentu dan kemudian diubah menjadi bilangan biner.



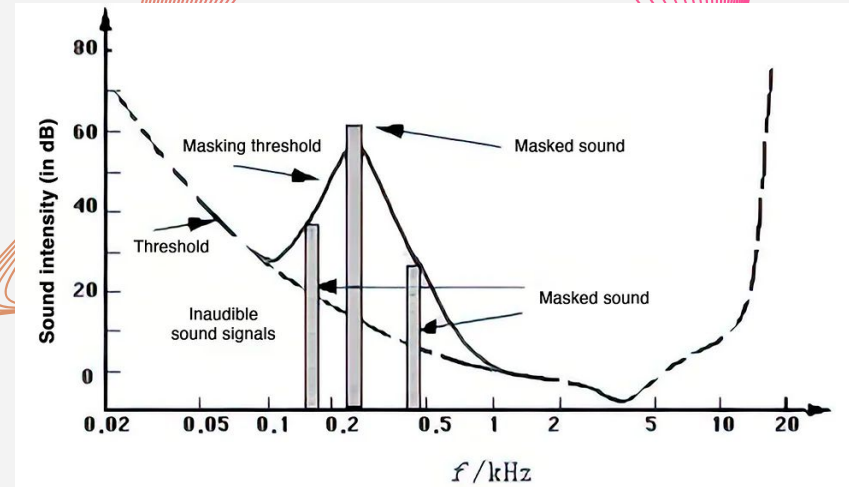
# Encoding

Setelah sinyal suara diubah menjadi sinyal digital, sinyal tersebut akan diubah menjadi bentuk yang lebih efisien. Proses ini disebut audio compression atau encoding. Terdapat beberapa metode yang digunakan dalam audio compression, seperti lossy dan lossless compression. Lossless compression akan mengurangi ukuran file tanpa mengurangi kualitas audio, sedangkan lossy compression akan mengurangi ukuran file dengan mengurangi kualitas audio. Beberapa metode lossy compression menghilangkan informasi yang dianggap tidak terlalu penting oleh pendengar manusia.



# Metode Encoding

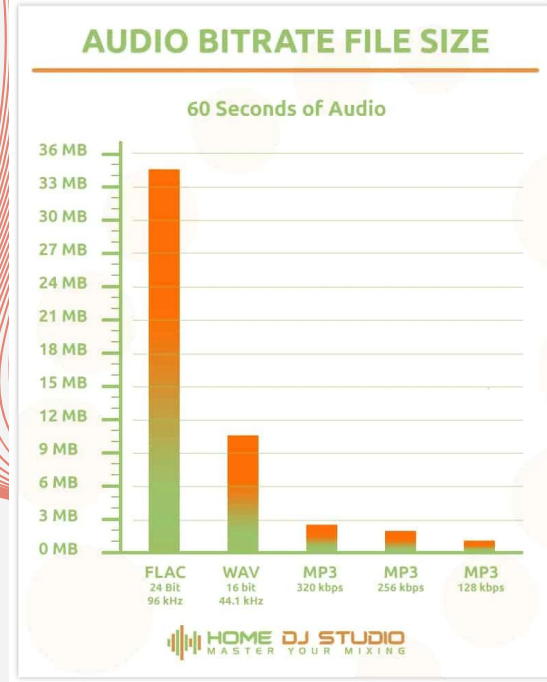
- Masking effect: Suara yang lebih keras akan menutupi suara yang lebih lemah.
- Frequency Selectivity: Manusia lebih sensitif terhadap perubahan frekuensi pada rentang (2-5Khz) dibandingkan dengan rentang frekuensi lainnya.
- Temporal masking: Suara yang terjadi setelah suara yang lebih keras akan diabaikan oleh pendengar.
- Perceptual coding: Menghilangkan informasi yang tidak terlalu penting bagi pendengar manusia (frekuensi di atas 20Khz, frekuensi di bawah 20Hz, dll).





# Bitrate Control

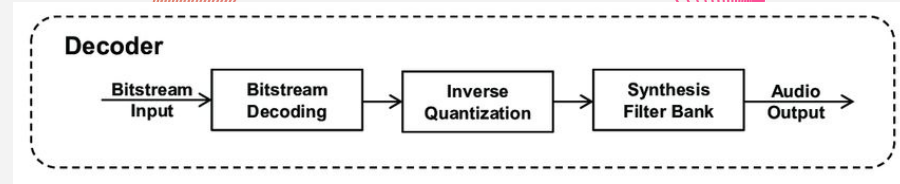
Bitrate control adalah proses mengatur seberapa banyak data yang akan digunakan untuk merepresentasikan sinyal audio. Bitrate yang lebih tinggi akan menghasilkan kualitas audio yang lebih baik, namun akan membutuhkan ruang penyimpanan yang lebih besar, dan juga sebaliknya.





# Audio Decoding

Setelah sinyal audio dikompresi, sinyal tersebut akan di-decode jika akan diputar atau digunakan.



# Mengapa Audio Coding Penting



Meningkatkan  
efisiensi  
penyimpanan  
data



Mengurangi  
penggunaan  
bandwidth saat  
mentransmisik  
an audio

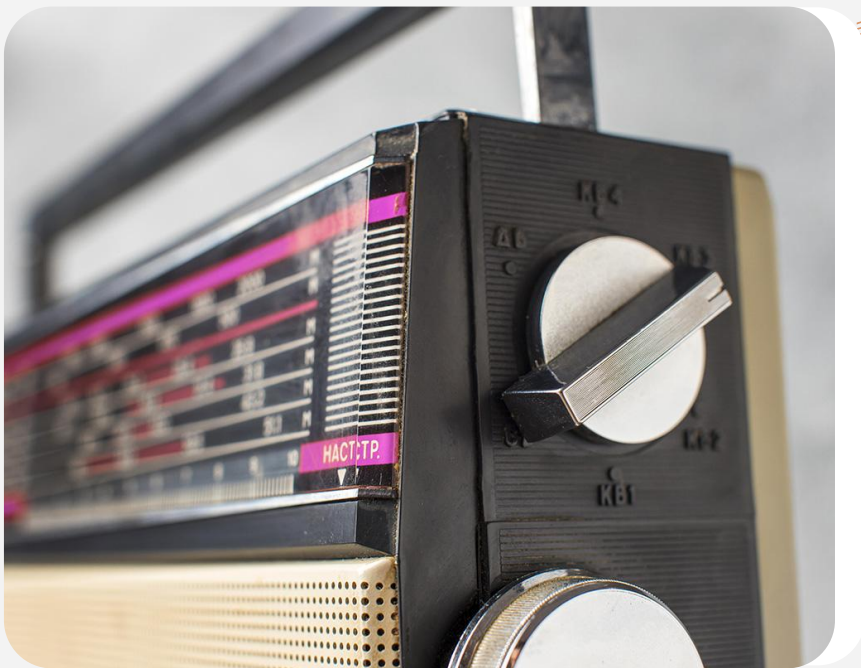


Meningkatkan  
Quality of  
Service (QoS)  
oleh pengguna



Menurunkan  
biaya  
infrastruktur  
jaringan





02

## Details of Audio Compression

# Faktor dalam Kualitas Audio Compression



Bit Rate



Sampling Rate



Bit Depth

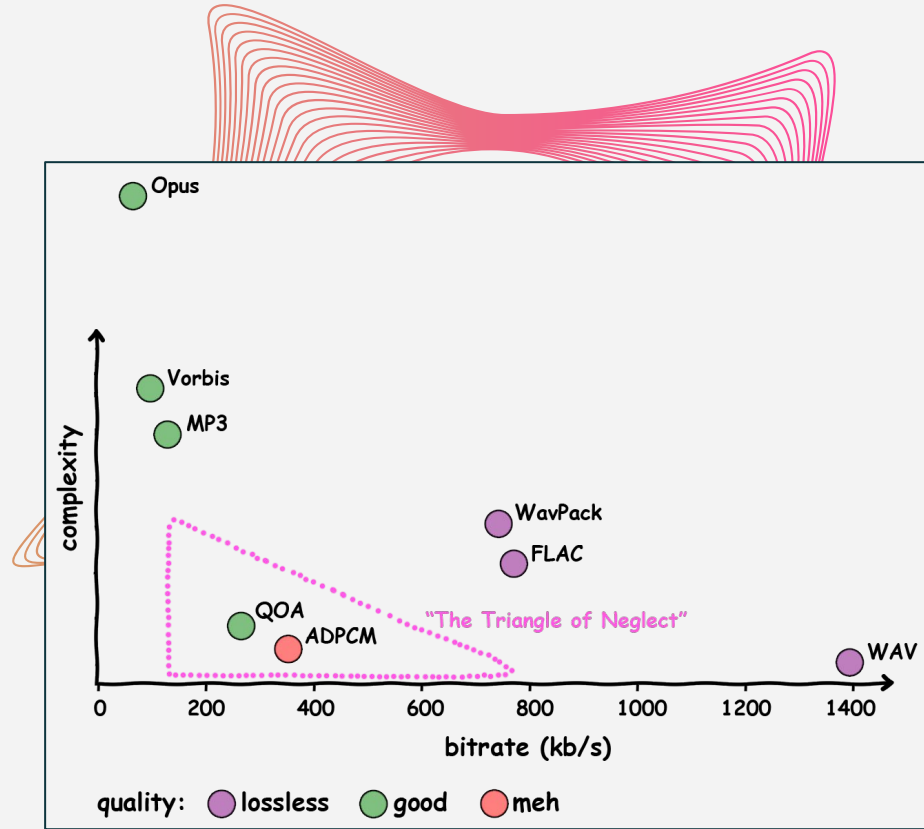


CODEC  
(Coder-Decoder)



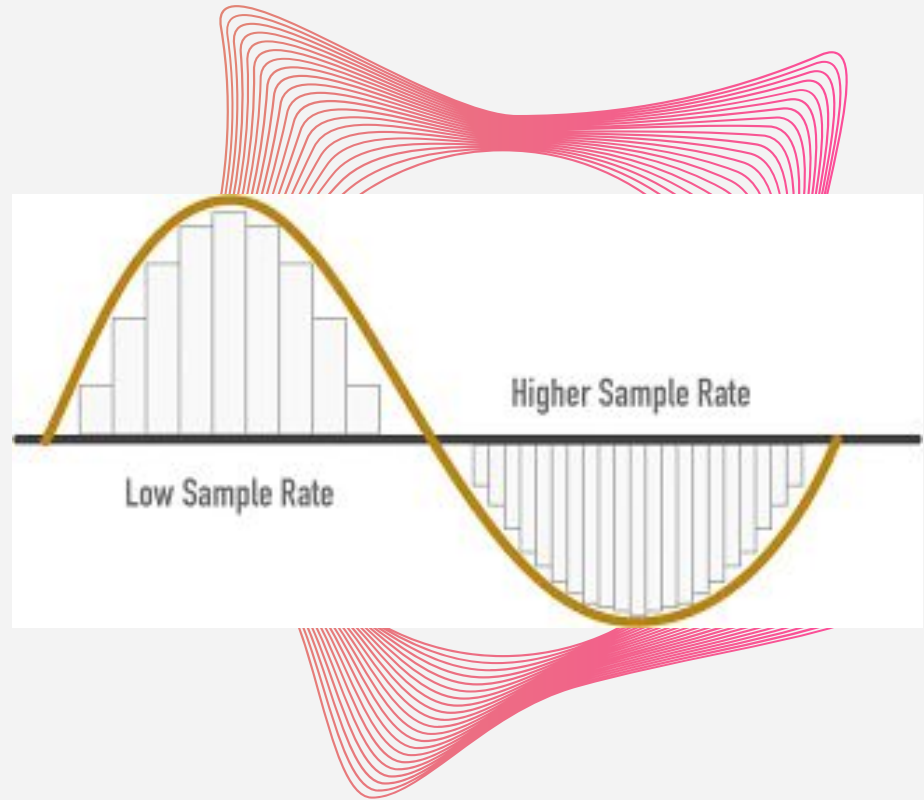
# Bit Rate

Bitrate adalah jumlah data yang digunakan untuk merepresentasikan sinyal audio dalam satu detik. Bitrate yang lebih tinggi akan menghasilkan kualitas audio yang lebih baik, namun akan membutuhkan ruang penyimpanan yang lebih besar.



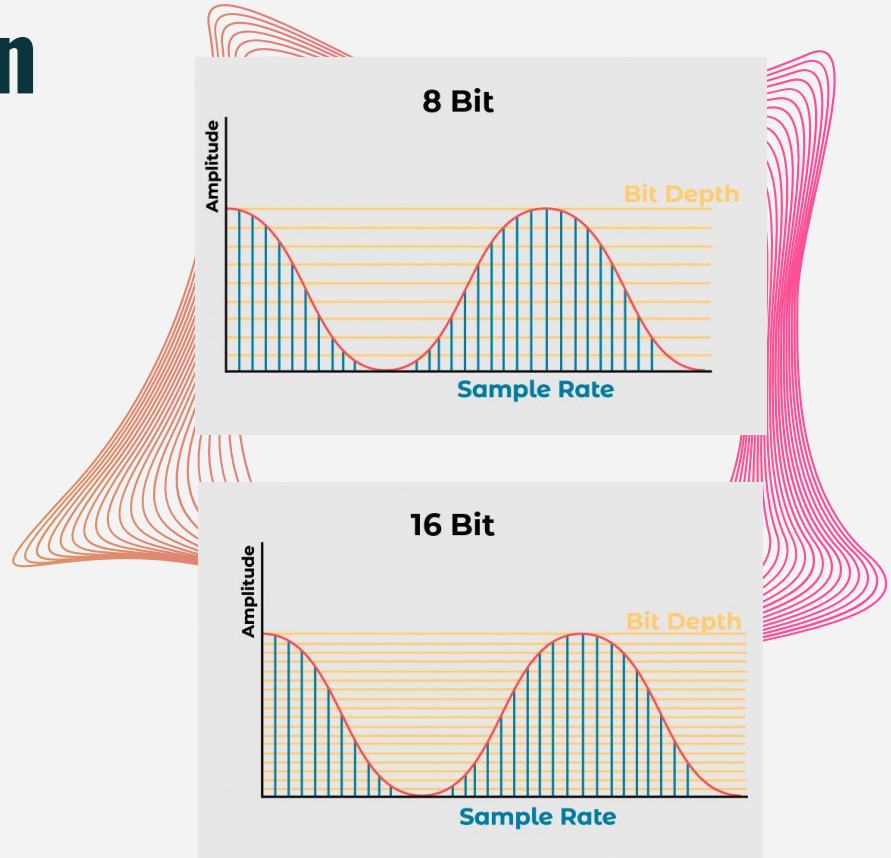
# Sampling Rate

Sampling rate adalah jumlah cuplikan yang diambil dalam satu detik. Sampling rate yang lebih tinggi akan menghasilkan kualitas audio yang lebih baik, namun akan membutuhkan ruang penyimpanan yang lebih besar.



# Bit Depth/Resolution

Bit depth adalah jumlah bit yang digunakan untuk merepresentasikan setiap cuplikan. Bit depth yang lebih tinggi akan menghasilkan kualitas audio yang lebih baik, namun akan membutuhkan ruang penyimpanan yang lebih besar.

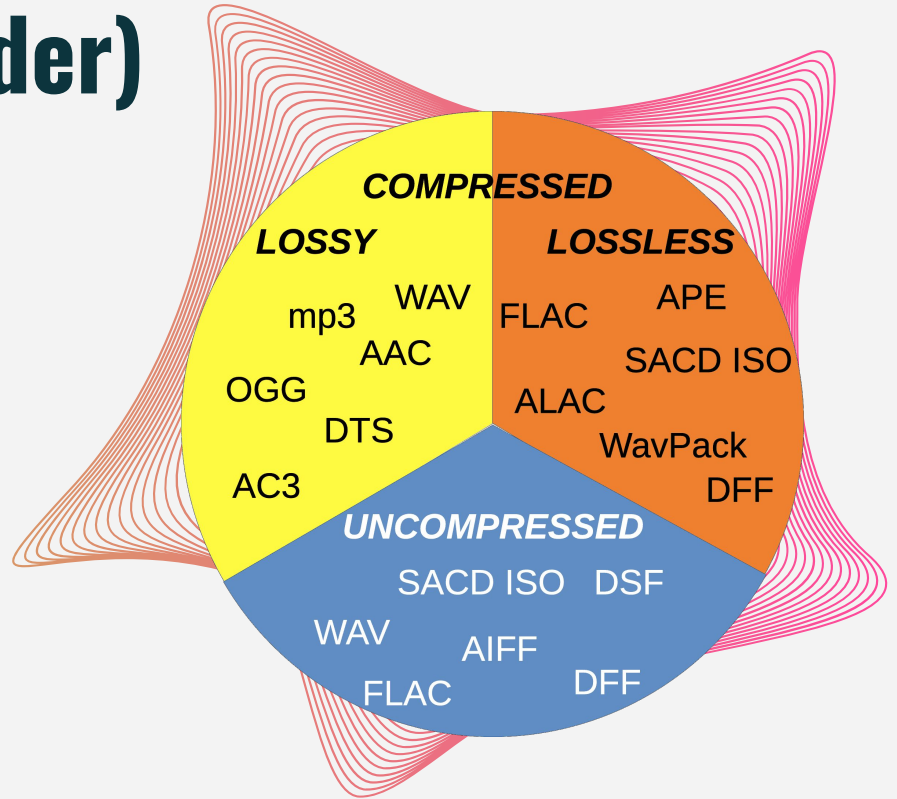




# CODEC (Coder-Decoder)

CODEC adalah algoritma yang digunakan untuk mengkompresi dan mendekompresi sinyal audio. CODEC yang lebih baik akan menghasilkan kualitas audio yang lebih baik, namun akan membutuhkan ruang penyimpanan yang lebih besar. Contoh CODEC yang sering digunakan dari urutan kualitas terbaik ke terburuk adalah:

- FLAC (Free Lossless Audio Codec)
- MP3 (MPEG-1 Audio Layer III)
- AAC (Advanced Audio Coding)



# Types of Audio Compression



Lossless  
Compression



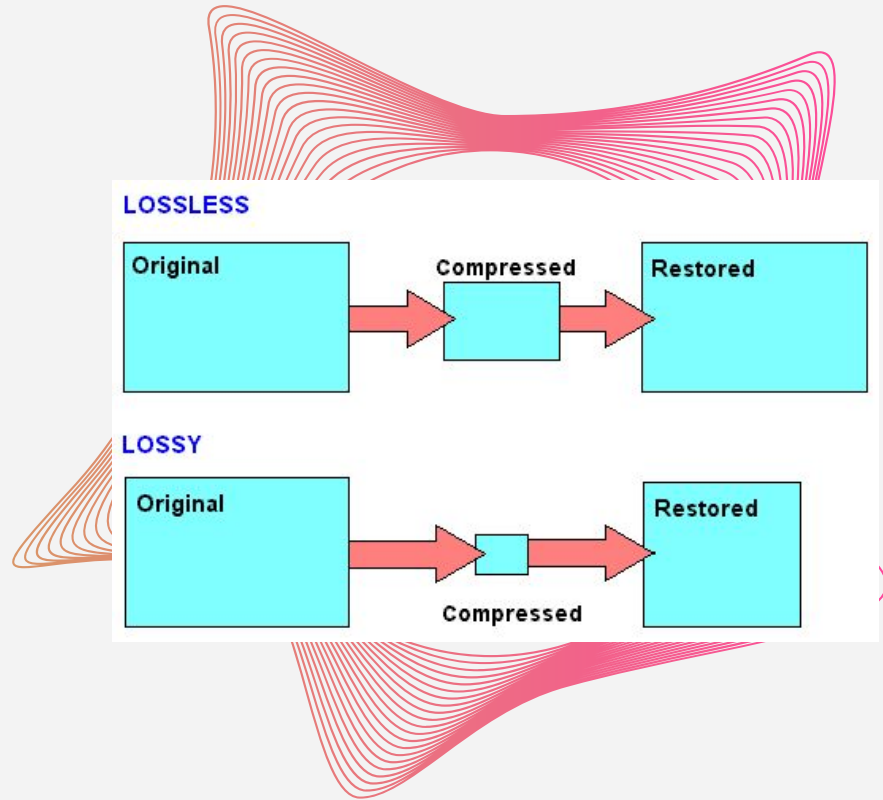
Lossy  
Compression



# Lossless Compression

Lossless compression adalah metode kompresi yang mengurangi ukuran file tanpa mengurangi kualitas audio. Lossless compression sering digunakan untuk menyimpan audio dengan kualitas yang sama dengan audio asli. Beberapa contoh lossless compression adalah:

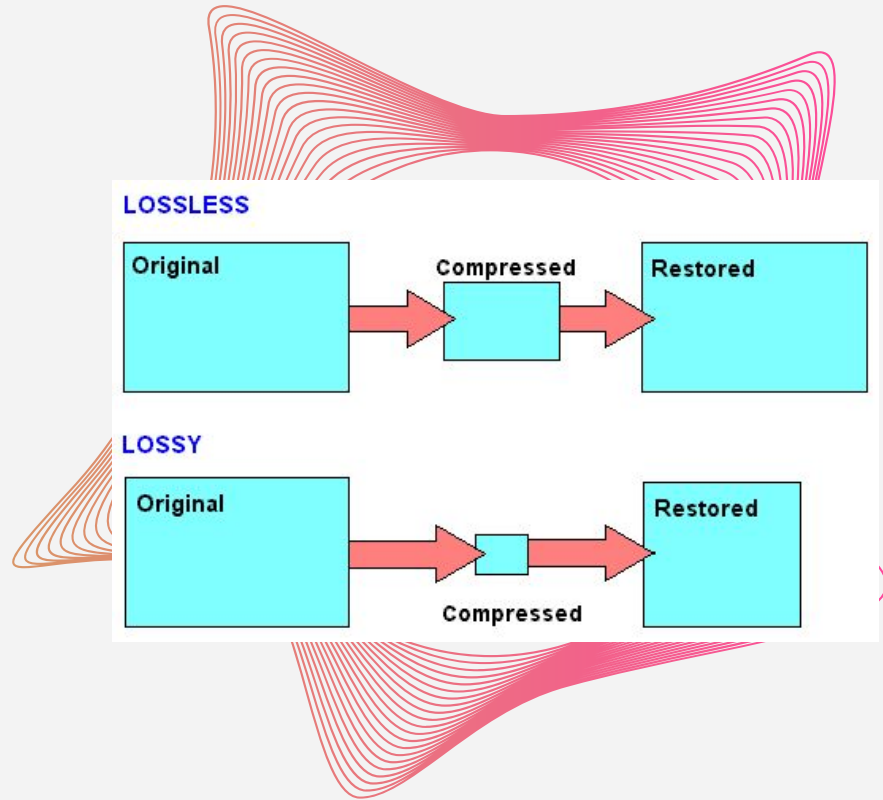
- FLAC (Free Lossless Audio Codec)
- ALAC (Apple Lossless Audio Codec)
- WAV (Waveform Audio File Format)

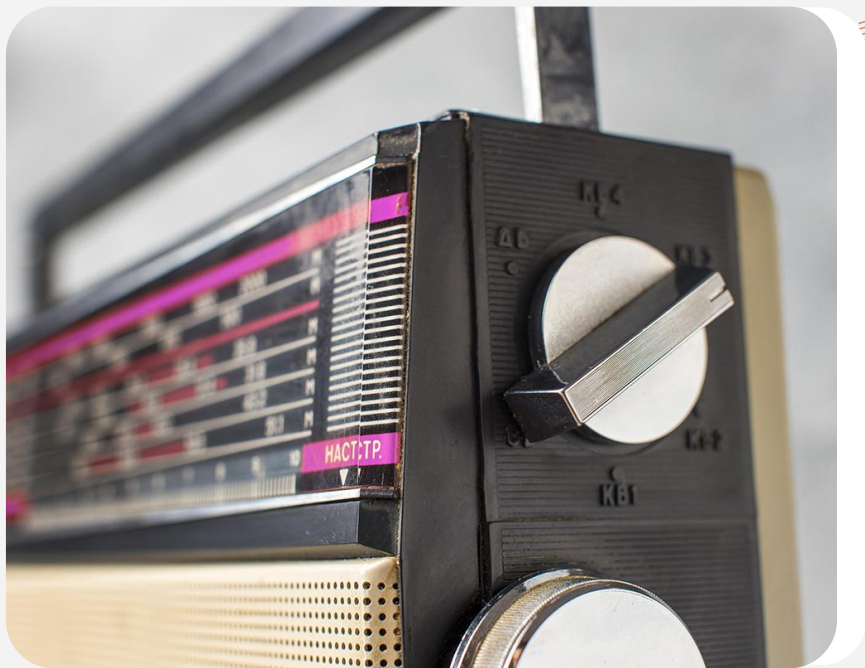


# Lossy Compression

Lossy compression adalah metode kompresi yang mengurangi ukuran file dengan mengurangi kualitas audio. Lossy compression sering digunakan untuk mentransmisikan audio melalui jaringan dengan kecepatan yang lebih tinggi atau memainkan audio dengan ekspektasi pengguna yang lebih rendah seperti audio streaming. Beberapa contoh lossy compression adalah:

- MP3 (MPEG-1 Audio Layer III)
- AAC (Advanced Audio Coding)
- OGG (Ogg Vorbis)





# 03

## Demo

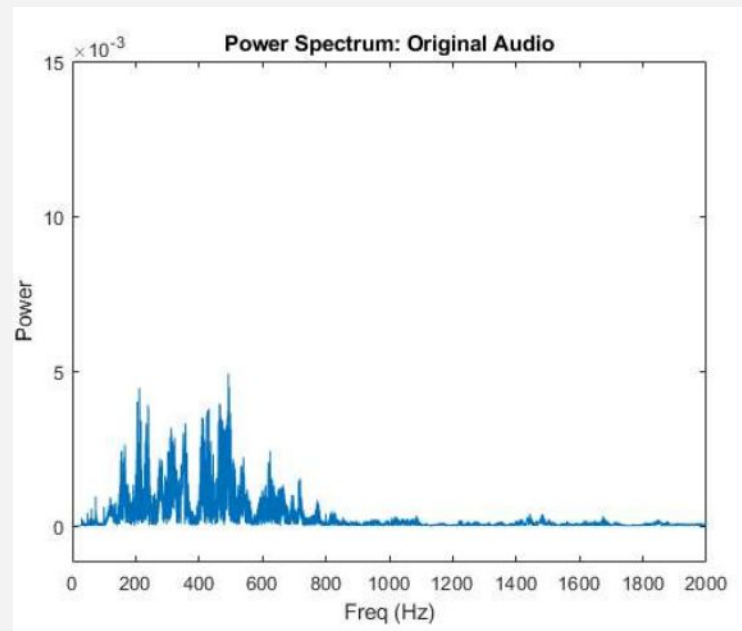
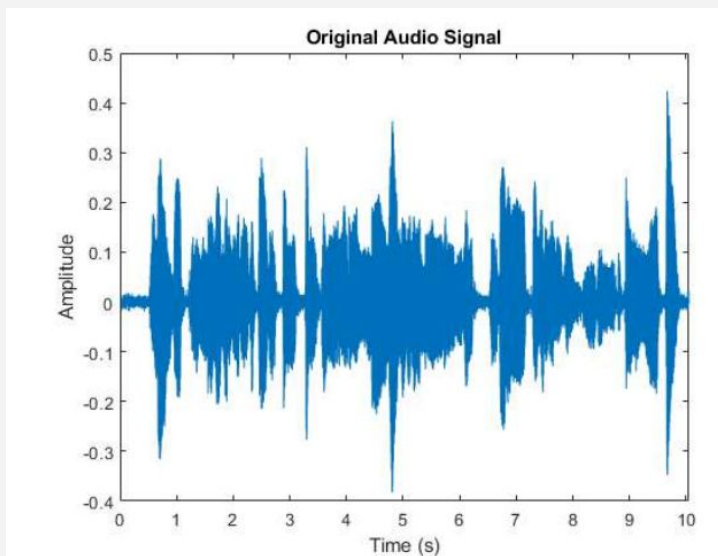
# Code

```
% Read and Play the audio file
myVoice = "C:\Users\chris\OneDrive\Documents\Kuliah\Semester 6\Multimedia\Week 2\voice_409422..
[x, fs] = audioread(myVoice);
sound(x,fs)
```

```
% Original Audio Signal Plotting
t = (0:1/fs:(length(x)-1)/fs);
plot (t,x)
title('Original Audio Signal');
xlabel("Time (s)")
ylabel("Amplitude")
xlim([0 t(end)])
```

```
% Compute and Plot the Power Spectrum
m = length(x);
n = pow2(nextpow2(m));
y = fft(x, n);
f = (0:n-1)*(fs/n);
power = 2/n * abs(y(1:n/2+1));
plot(f(1:floor(n/2)), power(1:floor(n/2)))
title('Power Spectrum: Original Audio');
xlabel("Freq (Hz)")
ylabel("Power")
xlim([0 2000])
ylim([-0.001171 0.015])
```

# Graph





# Code

```
% 1
N=length (x);
```

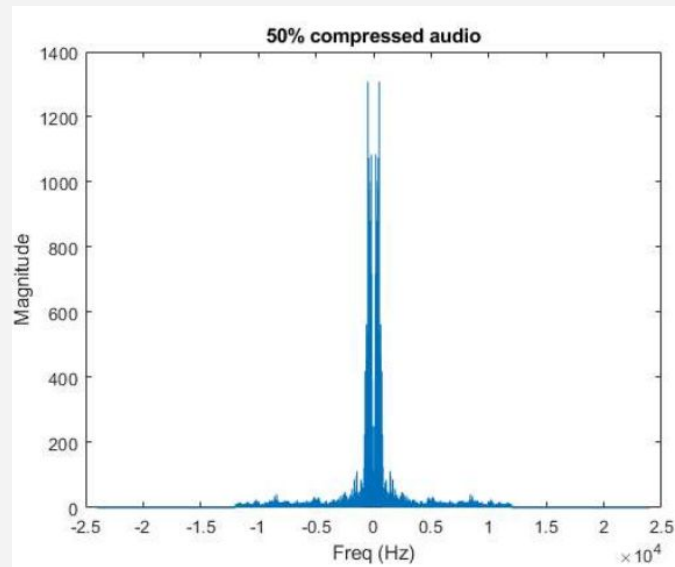
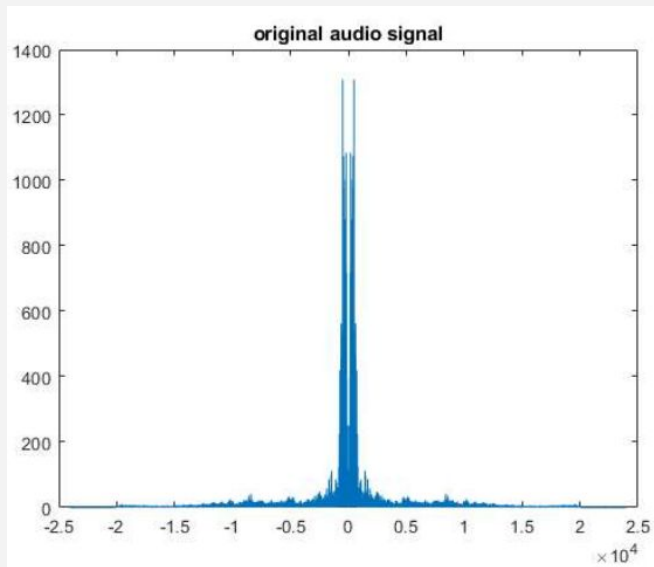
```
% 2
tfft=fft(x, N) ;
X=fftshift(tfft);
f =-fs/2:fs/N:(fs/2-fs/N);
plot(f,abs(X))
title ('original audio signal')
```

```
%3 compression
%Xr = zeros(1,N);
%Xr((N/4)+1:(3*N/4)) = X((N/4)+1:(3*N/4));
Xr = zeros(1, N);
Xr(round(N/4)+1 : round(3*N/4)) = X(round(N/4)+1 : round(3*N/4));
```

```
xr = real(ifft(fftshift(Xr))); %Reconstruction
audiowrite('50%compressed.wav',xr,fs);

plot(f,abs((Xr)));
title('50% compressed audio')
xlabel('Freq (Hz)'); ylabel('Magnitude');
```

# Graph



# Code

```
% Read and Play the compressed audio file
myVoice = "50%compressed.wav";
[x, fs] = audioread(myVoice);
sound(x,fs)
```

```
% Compressed Audio Signal Plotting
t = (0:1/fs:(length(x)-1)/fs);
plot (t,x)
title('Compressed Audio Signal');
xlabel("Time (s)")
ylabel("Amplitude")
xlim([0 t(end)])
```

```
% Compute and Plot the Power Spectrum
m = length(x);
n = pow2(nextpow2(m));
y = fft(x, n);
f = (0:n-1)*(fs/n);
power = 2/n * abs(y(1:n/2+1));
plot(f(1:floor(n/2)), power(1:floor(n/2)))
title('Power Spectrum: Compressed Audio');
xlabel("Freq (Hz)")
ylabel("Power")
xlim([0 2000])
ylim([-0.001171 0.015])
```

# Graph

