

## LAPORAN PRAKTIKUM

# OPERASI DASAR PENGOLAHAN SINYAL PADA DSP

Muhammad Fadhil Syahputra, Ismail Ragi Alfarugi .

Universitas Lambung Mangkurat

Email korespondensi : syahputraf112@gmail.com

## PENDAHULUAN

Pengolahan sinyal digital (Digital Signal Processing/DSP) merupakan bidang penting dalam teknik elektro dan informatika yang berfokus pada analisis, modifikasi, dan sintesis sinyal menggunakan sistem digital. Dengan semakin meluasnya penggunaan perangkat digital dalam berbagai aspek kehidupan—seperti komunikasi, audio, citra, dan kontrol sistem—pemahaman terhadap prinsip dasar DSP menjadi sangat krusial. Salah satu fondasi dari DSP adalah operasi-operasi dasar pengolahan sinyal, yang meliputi konvolusi, korelasi, transformasi Fourier, filter digital, dan lainnya. Operasi-operasi ini memungkinkan sinyal digital dianalisis dan dimanipulasi untuk berbagai keperluan, seperti peningkatan kualitas, ekstraksi informasi, hingga kompresi data.

Dalam praktiknya, DSP memberikan keunggulan dalam hal keakuratan, efisiensi, dan fleksibilitas dibandingkan dengan pengolahan sinyal analog. Oleh karena itu, mempelajari operasi dasar DSP tidak hanya menjadi bagian dari kurikulum akademik, tetapi juga merupakan fondasi penting bagi pengembangan teknologi yang lebih canggih di era digital saat ini.

## TINJAUAN PUSTAKA

Sinyal adalah arus data dalam suatu sistem yang mengalir dari jalur transmisi berisi tegangan, arus, atau kuat medan listrik yang mengkodekan informasi. Sinyal sebagai besaran fisik yang berubah-ubah menurut waktu ( $t$ ), ruang, atau variabel-variabel lainnya. Peranan sinyal salah satunya adalah mendukung kinerja alat komunikasi (Khairunnisa, 2019).

Jenis sinyal dengan nilai bilangan biner berbentuk digital disebut dengan sinyal digital atau sinyal waktu-diskrit. Pengiriman

informasi dengan kecepatan tinggi, berulang, dan masih mempertahankan kualitas dan kuantitas informasi adalah kelebihan dari sinyal digital. Karakteristik dari sinyal ini adalah dapat mengalami perubahan atau modifikasi secara tiba-tiba dan mempunyai besaran antara 0 dan 1. Informasi sinyal digital dapat berupa arus listrik, intensitas, fase atau polarisasi medan optik, atau elektromagnetik lainnya. Sinyal digital memiliki jenis gelombang kotak putus-putus dan hanya memiliki dua nilai amplitudo seiring dengan bertambahnya waktu. Alat elektronik seperti komputasi dan transmisi data umumnya menggunakan sinyal digital (Cindy et al., 2025).

Pemrosesan sinyal digital (DSP) adalah kegiatan untuk modifikasi, menganalisis, dan interpretasi sinyal digital di berbagai hal seperti komunikasi dan pemrosesan audio. DSP melibatkan penggunaan teknik matematika dan komputasi untuk memproses sinyal. Teknik pemodelan sinyal seperti filter Wiener, estimasi spektrum daya, dan penyaringan adaptif berperan penting untuk menganalisis dan memprediksi sinyal. Filter sinyal digital dalam perangkat lunak diimplementasikan dalam istilah bahasa tingkat tinggi, seperti C++ (Munir 2006). Sinyal digital dapat diaplikasikan untuk proses penyisipan informasi ke dalam data digital seperti citra, audio dan video. Discrete Wavelet Transform (DWT) adalah metode untuk menyisipkan informasi ke dalam citra, sedangkan Haar Wavelet adalah teknik pemrosesan sinyal digital menggunakan konsep citra yang ditransformasi terlebih dulu didekomposisi sub-sub citra sesuai dengan tujuan level transformasi (Utami et al., 2022).

## METODE PENELITIAN

### *Alat dan Bahan*

1. PC atau Laptop
2. Python 3.x
3. Pustaka python

### *Pembangkitan Sinyal*

#### *Pembangkitan Sinyal Persegi*

Sinyal persegi (square wave) adalah jenis sinyal diskrit atau kontinu yang hanya memiliki dua level amplitudo, biasanya berupa nilai tinggi (high) dan rendah (low), yang bergantian secara periodik. Sinyal ini banyak digunakan dalam sistem digital, pemicu (trigger), dan simulasi sistem switching karena karakteristiknya yang menyerupai sinyal biner. Pembangkitan sinyal persegi dapat dilakukan menggunakan fungsi logika atau metode matematika seperti penggunaan fungsi sign dari gelombang sinus. Dalam pengolahan sinyal digital, sinyal persegi menjadi dasar dalam pengujian sistem respons, analisis frekuensi, dan pembuatan sinyal uji untuk filter.

#### *Pembangkitan Sinyal Diskrit Sekuens Konstan*

Sinyal diskrit sekuens konstan adalah sinyal dengan nilai tetap (konstan) untuk setiap indeks waktu diskrit. Sinyal ini sering digunakan sebagai sinyal dasar dalam analisis sistem linier waktu diskrit, serta sebagai masukan pengujian untuk mengetahui kestabilan atau karakteristik sistem. Nilai konstan bisa berupa bilangan real atau kompleks, dan sinyal ini berguna untuk menyederhanakan perhitungan dalam simulasi digital.

#### *Pembangkitan Sinyal Waktu Diskrit Sekuens Pulsa*

Sinyal sekuens pulsa waktu diskrit (impulse/discrete delta function) adalah sinyal yang memiliki nilai 1 pada satu titik waktu tertentu (biasanya  $n=0$  atau  $n=0$ ) dan nol di waktu lainnya. Ini merupakan sinyal penting dalam teori sistem digital karena berperan sebagai identitas dalam konvolusi dan sebagai respon dasar dalam sistem linier. Pembangkitan sekuens pulsa membantu dalam menguji dan memahami respon impuls dari suatu sistem digital.

#### *Pembentukan Sinyal Sinus Waktu Diskrit*

Sinyal sinus waktu diskrit adalah representasi digital dari sinyal sinusoidal kontinu, yang dibentuk dengan sampling terhadap fungsi sinus pada interval waktu tertentu. Sinyal ini digunakan dalam berbagai

aplikasi seperti pemrosesan audio, komunikasi, dan analisis frekuensi. Bentuk umum dari sinyal ini adalah.

#### *Pembentukan Sinyal Audio Dengan File Wav*

Pembentukan sinyal audio dari file WAV melibatkan pembacaan data digital dari file audio berformat WAV, yang biasanya disampling pada frekuensi tertentu (misalnya 44.1 kHz). Data ini kemudian diolah sebagai sinyal diskrit untuk berbagai keperluan, seperti analisis spektral, pemfilteran, atau manipulasi audio. Proses ini penting dalam bidang pengolahan suara, di mana file audio dikonversi menjadi representasi numerik untuk keperluan analisis dan rekayasa.

### *Operasi Dasar Sinyal*

Operasi dasar sinyal adalah serangkaian transformasi atau manipulasi yang dilakukan terhadap sinyal untuk menganalisis, memodifikasi, atau mengoptimalkan penggunaannya dalam berbagai sistem. Operasi-operasi ini meliputi pergeseran waktu, pencerminan, penguatan, pelemahan, penjumlahan, perkalian, hingga penambahan noise. Dalam ranah pengolahan sinyal digital (DSP), operasi-operasi ini penting untuk memahami perilaku sistem terhadap masukan yang berbeda, membentuk sinyal baru dari sinyal eksisting, atau menyesuaikan sinyal dengan karakteristik sistem yang dituju.

Melalui operasi dasar ini, sinyal dapat direkayasa agar sesuai dengan kebutuhan aplikasi seperti komunikasi, audio, kontrol sistem, hingga kecerdasan buatan. Pemahaman terhadap operasi dasar ini menjadi fondasi penting dalam desain dan analisis sistem pengolahan sinyal digital.

#### *Pergeseran Sinyal*

Pergeseran sinyal (time shifting) adalah operasi memindahkan sinyal ke arah maju atau mundur dalam domain waktu. Jika  $x[n]$  digeser menjadi  $x[n-n_0]$ , maka sinyal bergeser ke kanan (delay), dan jika menjadi  $x[n+n_0]$ , maka sinyal bergeser ke kiri (advance). Operasi ini digunakan untuk mensimulasikan keterlambatan sinyal atau menyelaraskan sinyal dalam analisis.

#### *Pencerminan Sinyal*

Pencerminan sinyal (time reversal) adalah operasi membalik sinyal terhadap sumbu vertikal waktu, menghasilkan  $x[-n]$ . Operasi ini sering digunakan dalam konvolusi,

terutama dalam sistem linier waktu diskrit, untuk menghitung respon sistem terhadap sinyal input.

#### Penguatan Sinyal

Penguatan sinyal (amplification) dilakukan dengan mengalikan sinyal dengan konstanta lebih besar dari satu. Jika  $y[n]=k \cdot x[n]$ , maka amplitudo sinyal meningkat. Operasi ini digunakan untuk meningkatkan kekuatan sinyal atau menyesuaikan level sinyal ke sistem tertentu.

#### Pelemahan Sinyal

Kebalikan dari penguatan, pelemahan sinyal (attenuation) dilakukan dengan mengalikan sinyal dengan konstanta antara 0 dan 1. Jika  $0 < k < 1$ , maka  $y[n]=k \cdot x[n]$ , menghasilkan sinyal dengan amplitudo yang lebih kecil. Ini berguna dalam pengendalian level sinyal agar tidak melebihi batas sistem.

#### Penjumlahan dua Sinyal

Penjumlahan dua sinyal dilakukan dengan menambahkan nilai amplitudo masing-masing sinyal pada titik waktu yang sama:  $y[n]=x_1[n]+x_2[n]$ .

Operasi ini berguna dalam penggabungan sinyal, superposisi, serta dalam proses rekonstruksi dan interferensi sinyal.

#### Perkalian dua Sinyal

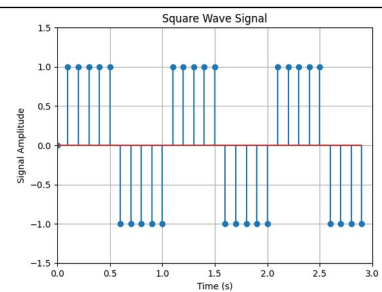
Perkalian dua sinyal (pointwise multiplication) adalah operasi mengalikan dua sinyal pada setiap titik waktu:  $y[n]=x_1[n] \cdot x_2[n]$ . Operasi ini sering digunakan dalam modulasi amplitudo, windowing, atau filter numerik berbasis waktu.

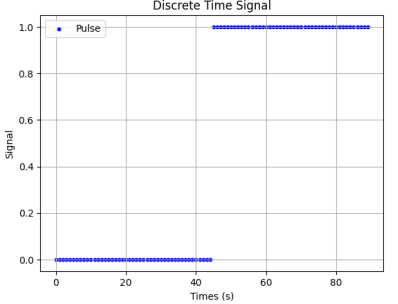
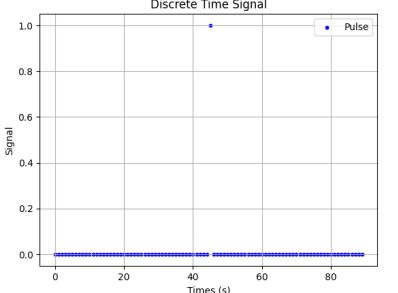
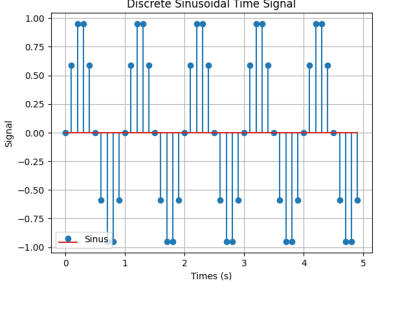
#### Penambahan Noise Gaussian pada Sinyal audio

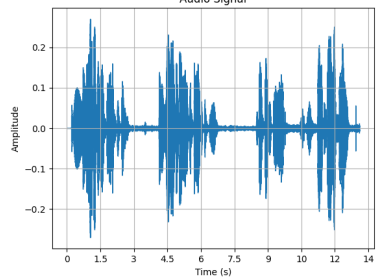
Penambahan noise Gaussian adalah proses menambahkan derau acak dengan distribusi normal (Gaussian) ke dalam sinyal audio untuk mensimulasikan kondisi nyata seperti gangguan transmisi atau pengujian ketahanan sistem terhadap noise. Noise ditambahkan ke sinyal sebagai  $y[n]=x[n]+\eta[n]$ , di mana  $\eta[n]$  adalah noise Gaussian. Teknik ini penting dalam pengujian sistem komunikasi dan pengolahan suara.

## HASIL DAN PEMBAHASAN

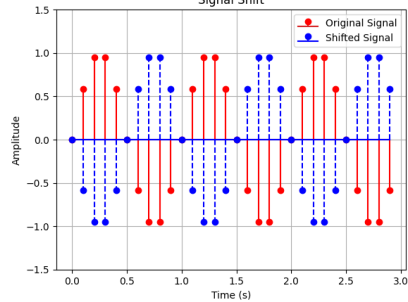
Tabel 1 Pembangkitan Sinyal

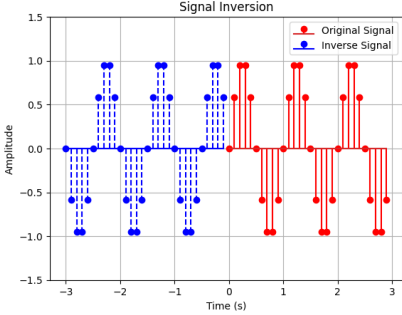
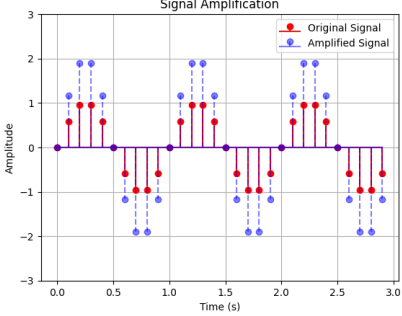
Judul Program	Kode Program	Hasil Program
# 1.4.1.2 Pembangkitan Sinyal Persegi	<pre># 1.4.1.2 Pembangkitan Sinyal Persegi analogFreq = 1 duration = 3 samplingFreq = 10 t = np.arange(0, duration, 1/samplingFreq) signal = np.sign(np.sin(2 * np.pi * analogFreq * t)) fig, ax = plt.subplots() ax.set_title('Square Wave Signal') ax.set_xlabel('Time (s)') ax.set_ylabel('Signal Amplitude') ax.set_xlim(0, duration) ax.set_ylim(-1.5, 1.5) ax.stem(t, signal) plt.grid() plt.show()</pre>	

#1.4.1.3 Pembangkitan Sinyal Waktu Diskrit, Sekuen Konstan	<pre># 1.4.1.3 Pembangkitan Sinyal Waktu Diskrit, Sekuen Konstan waveLength = 90 pulsePos = 45  t = np.arange(0, waveLength, 1) pulse = np.zeros(len(t))  for i in range(len(t)):     if t[i] &gt;= pulsePos:         pulse[i] = 1  fig, ax = plt.subplots() ax.scatter(t, pulse, label='Pulse',            color='blue', s=10) ax.set_xlabel('Times (s)') ax.set_ylabel('Signal') ax.set_title('Discrete Time Signal') ax.legend() plt.grid() plt.show()</pre>	
#1.4.1.4 Pembangkitan Sinyal Waktu Diskrit, Sekuen Pulsa	<pre># 1.4.1.4 Pembangkitan Sinyal Waktu Diskrit, Sekuen Pulsa  for i in range(len(t)):     if t[i] == pulsePos:         pulse[i] = 1     else:         pulse[i] = 0 fig, ax = plt.subplots() ax.scatter(t, pulse, label='Pulse',            color='blue', s=10) ax.set_xlabel('Times (s)') ax.set_ylabel('Signal') ax.set_title('Discrete Time Signal') ax.legend() plt.grid() plt.show()</pre>	
#1.4.1.5 Pembentukan Sinyal Sinus Waktu Diskrit	<pre># 1.4.1.5 Pembentukan Sinyal Sinus Waktu Diskrit analogFreq = 1 SamplingFreq = 10 t = np.arange(0, 5, 1/SamplingFreq) sinus = np.sin(2 * np.pi * analogFreq * t) fig, ax = plt.subplots() ax.stem(t, sinus, label='Sinus') ax.set_xlabel('Times (s)')</pre>	

	<pre>ax.set_ylabel('Signal') ax.set_title('Discrete Sinusoidal Time Signal') ax.legend() plt.grid() plt.show()</pre>	
# 1.4.1.6 Pembangkitan Sinyal Audio Dengan Memanfaatkan file *.wav	<pre># 1.4.1.6 Pembangkitan Sinyal Audio Dengan Memanfaatkan file *.wav y, sr = librosa.load('./audio.mp3', sr=None) # Load the audio file fig, ax = plt.subplots() librosa.display.waveshow(y, sr=sr, ax=ax) ax.set_xlabel('Time (s)') ax.set_ylabel('Amplitude') ax.set_title('Audio Signal') plt.grid() plt.show()</pre>	

Tabel 2 Operasi dasar Sinyal

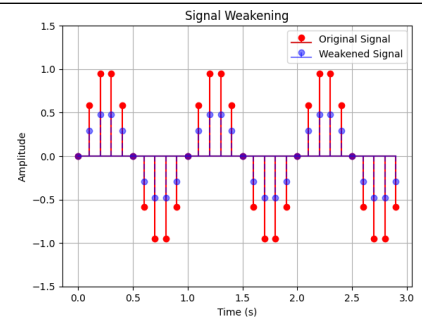
# 1.4.5.1 Pergeseran Sinyal	<pre># 1.4.5.1 Pergeseran Sinyal analogFreq = 1 samplingFreq = 10 duration = 3 t = np.arange(0, duration, 1/samplingFreq) signal = np.sin(2 * np.pi * analogFreq * t) shifted_signal = np.sin(2 * np.pi * analogFreq * (t - 1/2))  fig, ax = plt.subplots() markerline, stemline, baseline = ax.stem(t, signal, label='Original Signal') stemline.set_color('red') markerline.set_color('red') markerline, stemline, baseline = ax.stem(t, shifted_signal, label='Shifted Signal', ) stemline.set_color('blue') markerline.set_color('blue') stemline.set_linestyle('dashed') baseline.set_color('blue') ax.set_xlabel('Time (s)') ax.set_ylabel('Amplitude') ax.set_title('Signal Shift') ax.set_ylim(-1.5, 1.5) ax.legend(loc='upper right') plt.grid()</pre>	
--------------------------------	---	---

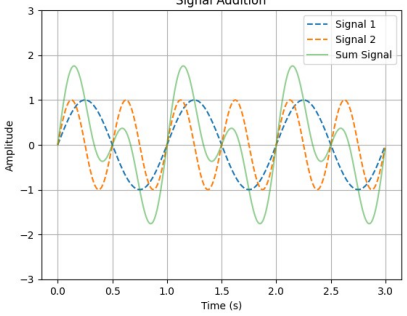
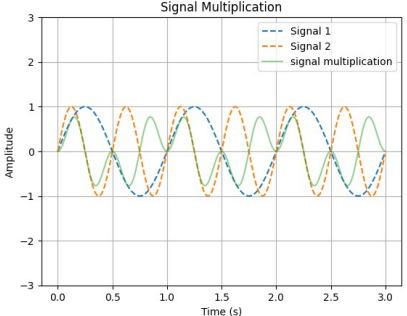
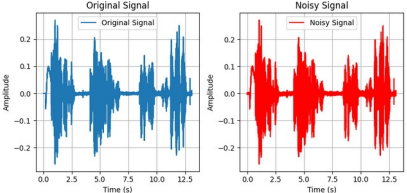
<p># 1.4.5.2 Pencerminan Sinyal</p>	<pre>plt.show()  # 1.4.5.2 Pencerminan Sinyal analogFreq = 1 samplingFreq = 10 duration = 3 oriT = np.arange(0, duration, 1/samplingFreq) inverseT = np.arange(-duration, 0, 1/samplingFreq) signal = np.sin(2 * np.pi * analogFreq * oriT) fig, ax = plt.subplots() markerline, stemline, baseline = ax.stem(oriT, signal, label='Original Signal') stemline.set_color('red') markerline.set_color('red') markerline, stemline, baseline = ax.stem(inverseT, signal*-1, label='Inverse Signal') stemline.set_color('blue') markerline.set_color('blue') stemline.set_linestyle('dashed') baseline.set_color('blue') ax.set_xlabel('Time (s)') ax.set_ylabel('Amplitude') ax.set_title('Signal Inversion') ax.set_ylim(-1.5, 1.5) ax.legend(loc='upper right') plt.grid() plt.show()</pre>	 <p>The plot titled 'Signal Inversion' shows two signals over time from -3 to 3 seconds. The 'Original Signal' (red solid line with dots) is a sine wave starting at 0. The 'Inverse Signal' (blue dashed line with dots) is an inverted sine wave starting at 0. The y-axis is labeled 'Amplitude' and ranges from -1.5 to 1.5. A legend in the upper right corner identifies the two signals.</p>
<p># 1.4.5.3 Penguatan Sinyal</p>	<pre># 1.4.5.3 Penguatan Sinyal Amplification = 2 analogFreq = 1 samplingFreq = 10 duration = 3 t = np.arange(0, duration, 1/samplingFreq) signal = np.sin(2 * np.pi * analogFreq * t) amplified_signal = Amplification * signal fig, ax = plt.subplots() markerline, stemline, baseline = ax.stem(t, signal, label='Original Signal') stemline.set_color('red') markerline.set_color('red') markerline, stemline, baseline = ax.stem(t, amplified_signal, label='Amplified Signal') stemline.set_color('blue')</pre>	 <p>The plot titled 'Signal Amplification' shows two signals over time from 0.0 to 3.0 seconds. The 'Original Signal' (red solid line with dots) is a sine wave with an amplitude of 1. The 'Amplified Signal' (blue dashed line with dots) is a sine wave with an amplitude of 2. The y-axis is labeled 'Amplitude' and ranges from -3 to 3. A legend in the upper right corner identifies the two signals.</p>

```
stemline.set_alpha(0.5)
markerline.set_color('blue')
markerline.set_alpha(0.5)
stemline.set_linestyle('dashed')
baseline.set_color('blue')
baseline.set_alpha(0.5)
ax.set_xlabel('Time (s)')
ax.set_ylabel('Amplitude')
ax.set_title('Signal Amplification')
ax.set_ylim(-3, 3)
ax.legend(loc='upper right')
plt.grid()
plt.show()
```

#1.4.5.4  
Pelemahan  
Sinyal

```
# 1.4.5.4 Pelemahan Sinyal
Amplification = 2
weakened = 1/Amplification
analogFreq = 1
samplingFreq = 10
duration = 3
t = np.arange(0, duration, 1/samplingFreq)
signal = np.sin(2 * np.pi * analogFreq * t)
amplified_signal = weakened * signal
fig, ax = plt.subplots()
markerline, stemline, baseline =
ax.stem(t, signal, label='Original
Signal')
stemline.set_color('red')
markerline.set_color('red')
markerline, stemline, baseline =
ax.stem(t, amplified_signal,
label='Weakened Signal')
stemline.set_color('blue')
stemline.set_alpha(0.5)
markerline.set_color('blue')
markerline.set_alpha(0.5)
stemline.set_linestyle('dashed')
baseline.set_color('blue')
baseline.set_alpha(0.5)
ax.set_xlabel('Time (s)')
ax.set_ylabel('Amplitude')
ax.set_title('Signal Weakening')
ax.set_ylim(-1.5, 1.5)
ax.legend(loc='upper right')
plt.grid()
plt.show()
```



#1.4.5.5 Penjumlahan Dua Sinyal	<pre># 1.4.5.5 Penjumlahan Dua Sinyal f1 = 1 f2 = 2 duration = 3 t = np.arange(0, duration, 1/200) signal1 = np.sin(2 * np.pi * f1 * t) signal2 = np.sin(2 * np.pi * f2 * t) signalsum = signal1 + signal2 fig, ax = plt.subplots() ax.plot(t, signal1, label='Signal 1', linestyle='dashed', alpha=1) ax.plot(t, signal2, label='Signal 2', linestyle='dashed', alpha=1) ax.plot(t, signalsum, label='Sum Signal', alpha=0.5) ax.set_xlabel('Time (s)') ax.set_ylabel('Amplitude') ax.set_title('Signal Addition') ax.set_ylim(-3, 3) ax.legend(loc='upper right') plt.grid() plt.show()</pre>	 <p>The plot titled 'Signal Addition' shows three sinusoidal waves over a time interval from 0.0 to 3.0 seconds. The y-axis represents 'Amplitude' ranging from -3 to 3. 'Signal 1' is a dashed blue line with an amplitude of 1. 'Signal 2' is a dashed orange line with an amplitude of 2. 'Sum Signal' is a solid green line representing the sum of the two signals, with an amplitude of 3. The plot includes a grid and a legend in the upper right corner.</p>
#1.4.5.6 Perkalian Dua Sinyal	<pre>f1 = 1 f2 = 2 duration = 3 t = np.arange(0, duration, 1/200) signal1 = np.sin(2 * np.pi * f1 * t) signal2 = np.sin(2 * np.pi * f2 * t) signalsum = signal1*signal2 fig, ax = plt.subplots() ax.plot(t, signal1, label='Signal 1', linestyle='dashed', alpha=1) ax.plot(t, signal2, label='Signal 2', linestyle='dashed', alpha=1) ax.plot(t, signalsum, label='signal multiplication', alpha=0.5) ax.set_xlabel('Time (s)') ax.set_ylabel('Amplitude') ax.set_title('Signal Multiplication') ax.set_ylim(-3, 3) ax.legend(loc='upper right') plt.grid() plt.show()</pre>	 <p>The plot titled 'Signal Multiplication' shows the same two input signals as the addition plot: 'Signal 1' (dashed blue) and 'Signal 2' (dashed orange). The third series, 'signal multiplication', is a solid green line representing the product of the two signals. The amplitude of the product signal varies between -1 and 1. The plot includes a grid and a legend in the upper right corner.</p>
#1.4.5.7 Penambahan Noise Gaussian pada	<pre># 1.4.5.7 Penambahan Noise Gaussian pada Sinyal Audio constant = 0.001 y, sr = librosa.load('./audio.mp3', sr=None)</pre>	 <p>This section contains two side-by-side plots. The left plot, titled 'Original Signal', shows a blue waveform representing the original audio signal over a time interval from 0.0 to 12.5 seconds, with an amplitude range of -0.2 to 0.2. The right plot, titled 'Noisy Signal', shows a red waveform representing the same audio signal with added Gaussian noise. The axes and scales are identical to the 'Original Signal' plot.</p>



<p>Sinyal Audio</p>	<pre> t = np.arange(0, len(y)) / sr noised_y = y + constant*np.random.normal(0, 1, len(y)) fig, ax = plt.subplots(1, 2, figsize=(8, 4)) ax[0].plot(t, y, label='Original Signal') ax[0].set_title('Original Signal') ax[0].set_xlabel('Time (s)') ax[0].set_ylabel('Amplitud') ax[0].legend() ax[0].grid() ax[1].plot(t, noised_y, label='Noisy Signal', color='red') ax[1].set_title('Noisy Signal') ax[1].set_xlabel('Time (s)') ax[1].set_ylabel('Amplitud') ax[1].legend() ax[1].grid() plt.tight_layout() plt.show() </pre>	
<p># 1.4.5.8 Proses Penguatan pada Sinyal Audio</p>	<pre> # 1.4.5.8 Proses Penguatan pada Sinyal Audio y, sr = librosa.load('./audio.mp3', sr=None) t = np.linspace(0, len(y)/sr, len(y)) Amplification = 2 amplified_y = Amplification * y fig, ax = plt.subplots(1, 2, figsize=(8, 4)) ax[0].plot(t, y, label='Original Signal') ax[0].set_title('Original Signal') ax[0].set_xlabel('Time (s)') ax[0].set_ylabel('Amplitud') ax[0].legend() ax[0].grid() ax[1].plot(t, amplified_y, label='Amplified Signal', color='red') ax[1].set_title('Amplified Signal') ax[1].set_xlabel('Time (s)') ax[1].set_ylabel('Amplitud') ax[1].legend() ax[1].grid() ax[0].set_ylim(ax[1].get_ylim()) plt.tight_layout() plt.show() </pre>	

#### PEMBAHASAN

Rangkaian kode ini menyajikan eksplorasi

komprehensif terhadap konsep dasar pengolahan sinyal digital, yang meliputi pembangkitan sinyal diskrit dan kontinu, transformasi dasar sinyal, serta manipulasi dan kombinasi sinyal dalam domain waktu. Pada bagian awal, dilakukan proses pembangkitan sinyal persegi (square wave) dengan memanfaatkan fungsi sinusoidal dari numpy yang dikombinasikan dengan fungsi tanda (`np.sign`) untuk menghasilkan nilai biner (+1 dan -1). Sinyal ini dibangkitkan dengan frekuensi analog sebesar 1 Hz dan frekuensi sampling sebesar 10 Hz, selama durasi tiga detik. Penggunaan `np.arange` memastikan bahwa waktu diskrit dibentuk dengan tepat, dan visualisasi menggunakan `stem plot` dari pustaka `matplotlib` memperlihatkan representasi sinyal dalam bentuk diskrit, yang umum dijumpai dalam sistem digital seperti mikrokontroler dan DSP (Digital Signal Processing).

Setelah itu, kode memperkenalkan pembangkitan sinyal waktu diskrit sederhana, yaitu sekuen konstan dan pulsa. Sekuen konstan dibentuk dengan memberikan nilai 1 pada semua titik waktu setelah indeks tertentu, sementara sekuen pulsa hanya memberikan nilai 1 pada satu titik waktu, dan sisanya tetap nol. Kedua jenis sinyal ini merupakan komponen dasar dalam sistem digital dan sering digunakan sebagai sinyal uji untuk mengevaluasi respon sistem atau filter. Visualisasi menggunakan `stem` juga digunakan untuk menggambarkan sifat diskrit dari sinyal-sinyal ini.

Selanjutnya, sinyal sinusoidal waktu diskrit dibangkitkan dengan frekuensi 1 Hz, menggunakan metode yang mirip dengan pembangkitan sinyal persegi sebelumnya, tetapi tanpa fungsi tanda. Sinyal ini divisualisasikan untuk memperlihatkan bentuk gelombangnya secara diskrit, yang penting dalam aplikasi seperti pemrosesan sinyal audio dan komunikasi digital. Bagian ini memperkuat pemahaman tentang perbedaan antara sinyal analog yang kontinu dan sinyal diskrit yang terdiri dari nilai-nilai pada waktu tertentu.

Bagian yang lebih kompleks dimulai dengan pembacaan sinyal audio eksternal (file .mp3) menggunakan pustaka `librosa`, yang sangat berguna dalam pengolahan sinyal audio berbasis Python. Sinyal audio yang berhasil dimuat divisualisasikan dalam domain waktu menggunakan `waveshow` dari `librosa.display`, yang menunjukkan amplitudo sinyal terhadap

waktu. Ini memberikan representasi nyata dari sinyal audio yang sebenarnya kita dengarkan, dan membuka pintu menuju pemrosesan sinyal suara, seperti pengenalan suara atau analisis musik.

Berbagai transformasi sinyal diterapkan untuk menggambarkan bagaimana sinyal dapat dimodifikasi. Transformasi pertama adalah pergeseran (time shifting), di mana sinyal digeser ke kanan (waktu ditunda) untuk menunjukkan efek delay atau latensi dalam sistem. Transformasi berikutnya adalah pencerminan (time reversal) dengan membalik urutan sinyal, yang penting untuk memahami operasi konvolusi. Sinyal kemudian diubah dengan amplifikasi (penguatan), yaitu dengan mengalikan seluruh amplitudo sinyal dengan faktor konstan (misalnya 2), yang berguna dalam sistem audio atau ketika sinyal terlalu lemah. Sebaliknya, pelemahan dilakukan dengan mengalikan sinyal dengan faktor kurang dari satu (misalnya 0.5) untuk mengecilkan amplitudo sinyal. Semua hasil transformasi tersebut dibandingkan dengan sinyal asli menggunakan plot berdampingan agar perbedaan efeknya dapat terlihat jelas.

Selanjutnya, dilakukan proses penjumlahan dua sinyal sinusoidal dengan frekuensi berbeda (1 Hz dan 2 Hz). Ini memperlihatkan bagaimana dua sinyal dapat digabungkan untuk membentuk sinyal baru yang lebih kompleks, dengan karakteristik frekuensi yang mencerminkan superposisi keduanya. Proses ini umum terjadi dalam sistem komunikasi ketika dua sinyal informasi dikombinasikan. Selain itu, perkalian dua sinyal sinusoidal juga dilakukan, yang menghasilkan sinyal dengan modulasi amplitudo — suatu konsep penting dalam teknik modulasi analog seperti AM (Amplitude Modulation). Visualisasi hasil penjumlahan dan perkalian ini menggambarkan interaksi frekuensi dalam sinyal dan menunjukkan pentingnya operasi aritmatika dalam domain waktu.

Pada bagian akhir, ditambahkan noise Gaussian ke dalam sinyal audio untuk mensimulasikan situasi nyata di mana sinyal tidak murni, melainkan tercampur dengan gangguan (noise), baik dari lingkungan maupun dari sistem elektronik itu sendiri. Noise ini dihasilkan dengan fungsi `np.random.normal`, yang menyediakan distribusi normal dengan rata-rata nol dan deviasi standar tertentu. Sinyal hasilnya

kemudian divisualisasikan dan dibandingkan dengan sinyal asli. Terakhir, sinyal audio juga diperkuat (gain) dengan mengalikan amplitudo dengan faktor tertentu untuk menunjukkan bagaimana sistem audio atau perangkat DSP mengatur volume sinyal.

Secara keseluruhan, seluruh kode ini mencerminkan berbagai aspek penting dalam pengolahan sinyal digital, dari pembangkitan, representasi, transformasi, hingga manipulasi sinyal baik diskrit maupun kontinu. Hal-hal yang dipelajari dari proses ini sangat esensial dalam bidang teknik elektro, informatika, sistem komunikasi, embedded systems, dan pemrosesan suara. Materi ini memberikan fondasi kuat untuk memahami bagaimana sinyal direpresentasikan secara digital, bagaimana mereka dapat dimanipulasi untuk keperluan analisis maupun transmisi, serta bagaimana sistem nyata dapat merespons berbagai bentuk sinyal yang masuk ke dalamnya.

#### **DAFTAR PUSTAKA**

- Cindy, Ulfah, N. N., Saragih, E., Sinaga, R. S. F., & Irwan. (2025). Pengolahan dan Pemrosesan Sinyal Digital. *Jurnal Media Informatika [JUMIN]*, 6(2), 1339-1344.
- Khairunnisa. (2019). *Pengolahan Sinyal*. Poliban Press, Yogyakarta.
- Munir, R. (2006). *Sekilas Image Watermarking untuk Memproteksi Citra Digital*. Institut Teknologi Bandung, Bandung.
- Utami, M., Rismawan, T., & Ristian, U. (2022). Implementasi Metode Discrete Wavelet Transform (DWT) pada Watermarking Citra Digital Keaslian Karya Berbasis WEB. *Coding : Jurnal Komputer dan Aplikasi*, 10(1), 124-135.