

# **Tugas 2**

## **AI-enabled Internet of Things - 2021-2**

Disusun untuk memenuhi tugas Mata Kuliah

IoT



Oleh:

Imam Ilham Khawarizma

1301180436

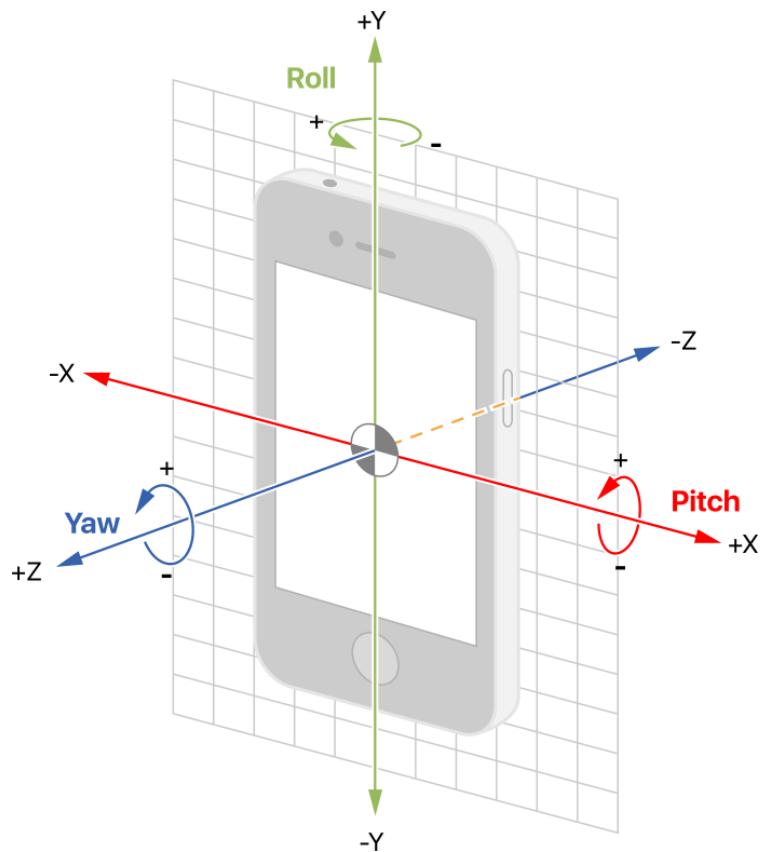
IF-42-GAB02

**Jurusan S1 Informatika**

**Fakultas Informatika**

**Telkom University**

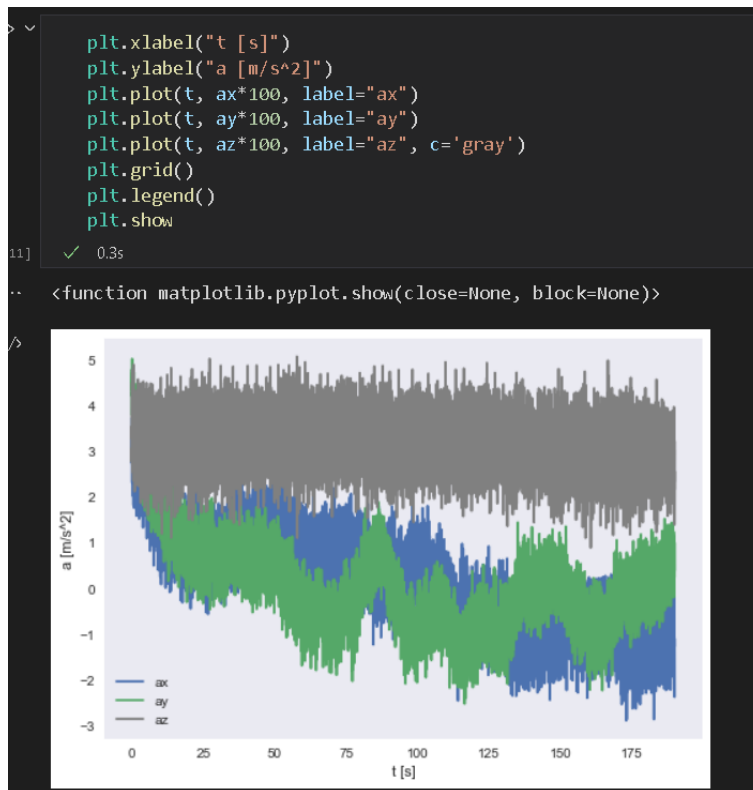
**2020**



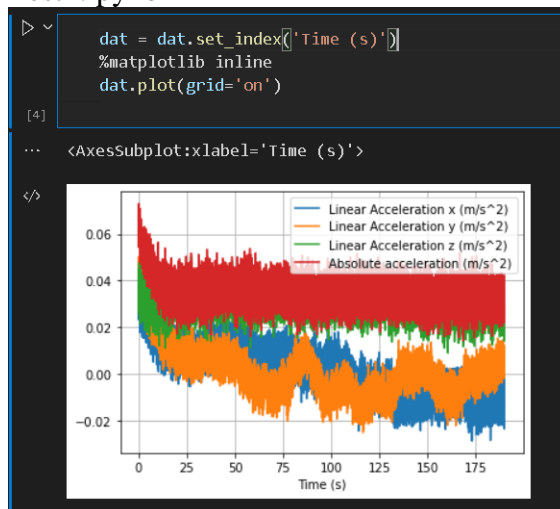
Dalam percobaan ini saya mengambil 3 sample data dan hanya menggunakan 1 sampel data. Sampel data yang saya gunakan menggunakan device iPhone 12 Pro, dan data yang diproses hanya sumbu X dan Y karena device sejajar dengan Y, bersinggungan dengan X, dan mengarah keatas +.Z

## Nomor 1.

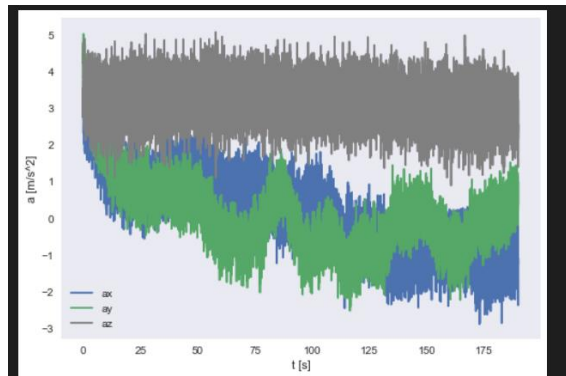
### a) Lampiran plot



## Test2.ipynb



Normal



Multiplied by 100 for magnification

## Test1.ipynb

### Penjelasan Umum:

Ini merupakan hasil plot dari perekaman yang telah dilakukan pada tanggal

18/04/2022..

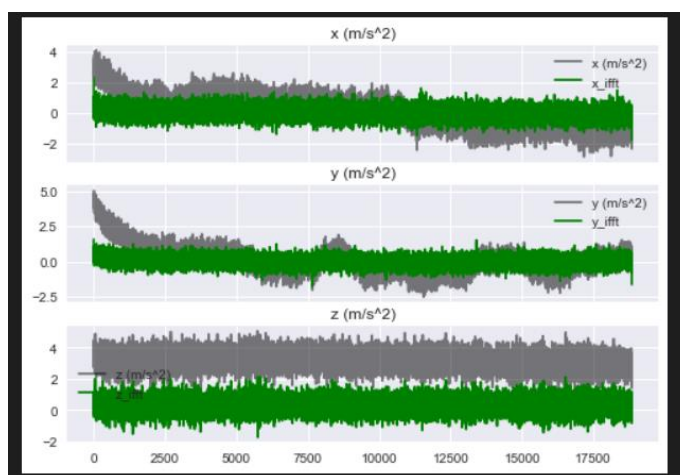
**b) Penjelasan komprehensif berdasarkan program yang dilampirkan:**

Program dibuat menggunakan python jupyter dengan library import pandas, numpy, matplotlib, dan math. Program dijalankan dari file .ipynb dan untuk menjalankan .xls digunakan pandas.

Dari hasil plotting, sumbu accelerometer yang dominan yakni sumbu x dan y berdasarkan apa yang terlihat diatas.

## Nomor 2

a) Ubahlah domain sumbu x dalam plot anda kedalam domain waktu (bukan index data) dengan cara melakukan proses FFT terhadap sinyal accelerometer yang anda dapatkan. Buatlah plot dari hasil FFT anda, serta jelaskan hasil dari FFT anda (misalnya jika tidak ada kendaraan lewat, spt apa frekuensi yg dihasilkan dalam doman FFT, jika ada truk lewat spt apa, dst.)



Untuk mengetahui apakah suatu kendaraan lewat atau tidak harus menggunakan pembandingan dan juga memerlukan instrumen tambahan supaya akurasi analisa lebih tepat. Bila melihat dari plot sumbu Y terdapat getaran

**b) Lampirkan serta jelaskan algoritma beserta kode program dari FFT yang anda gunakan !**

```

# Try to remove noise via Fourier analysis
# Discrete Fourier Transform sample frequencies
freq = np.fft.rfftfreq(df['Linear Acceleration x (m/s^2)'].size,d=dt)

# Compute the Fast Fourier Transform (FFT) of acceleration signals
fft_x = np.fft.rfft(df['Linear Acceleration x (m/s^2)'])
fft_y = np.fft.rfft(df['Linear Acceleration y (m/s^2)'])
fft_z = np.fft.rfft(df['Linear Acceleration z (m/s^2)'])

# Plot Frequency spectrum
fig4,[ax1,ax2,ax3] = plt.subplots(3,1,sharex=True,sharey=True)
fig4.suptitle('Noise Spectrum',fontsize=20)
ax1.plot(freq,abs(fft_x),c='r',label='x noise')
ax1.legend()
ax2.plot(freq,abs(fft_y),c='b',label='y noise')
ax2.legend()
ax3.plot(freq,abs(fft_z),c='g',label='z noise')
ax3.legend()
ax3.set_xlabel('Frequency (Hz)')
plt.show()
✓ 0.3s

```

FFT atau Fast Fourier Transform. Analisis Fourier mengubah sinyal dari domain aslinya (seringkali ruang atau waktu) menjadi representasi dalam domain frekuensi dan sebaliknya.

Pada awalnya mengidentifikasi frequency terlebih dahulu dengan sampel sumbu x. Lalu setiap sumbu dihitung dengan metode FFT menggunakan fft numpy library. Proses akhir melakukan plot FFT.

```

# Attenuate noise in X,Y,Z below 10Hz by 10 dB
atten_x_fft = np.where(freq < 10,fft_x * 0.1, fft_x)
atten_y_fft = np.where(freq < 10,fft_y * 0.1, fft_y)
atten_z_fft = np.where(freq < 10,fft_z * 0.1, fft_z)
print(atten_x_fft)

# Compute inverse of discrete Fourier Transform
df['x_iftt'] = np.fft.irfft(atten_x_fft,n=df.shape[0])
df['y_iftt'] = np.fft.irfft(atten_y_fft,n=df.shape[0])
df['z_iftt'] = np.fft.irfft(atten_z_fft,n=df.shape[0])

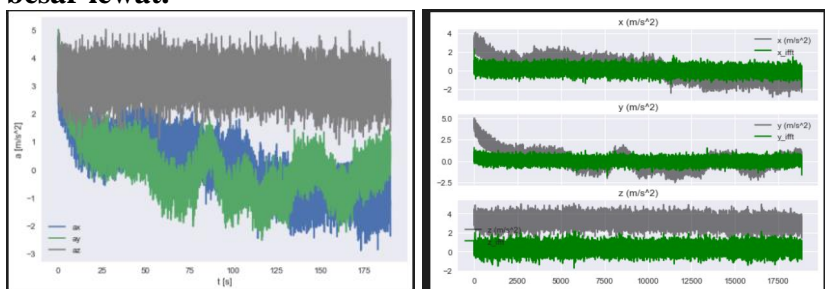
# Plot new acceleration signals
cols_raw = ['Linear Acceleration x (m/s^2)','Linear Acceleration y (m/s^2)','Linear Acceleration z (m/s^2)']
cols_new = ['x_iftt','y_iftt','z_iftt']
rawplot = df.plot( y=cols_raw, subplots=True, sharex=True,style='k',title=cols_raw,alpha=0.5 )
df.plot(y=cols_new,subplots=True,layout=(1,3),ax=rawplot,
sharex=True,style='g')
plt.show()
✓ 1.7s

```

Lalu setelah itu melakukan filtering noise, inverse FFT serta melakukan plotnya

c) Pada bagian ini anda akan melakukan analisis berdasarkan proses FFT untuk mengetahui frekuensi harmonik berdasarkan getaran tiap jenis kendaraan.

i. Berdasarkan plot data accelerometer, bedakan dengan cara memberi tanda secara manual, di saat kapan kendaraan kecil lewat, atau kendaraan sedang dan besar lewat.



Dengan menggunakan proses FFT akan lebih mudah mengetahui kendaraan lewat

dibanding hanya menggunakan raw data. Namun untuk mengetahui apakah suatu kendaraan lewat atau tidak harus menggunakan pembanding dan juga memerlukan instrumen tambahan supaya akurasi analisa lebih tepat. Berdasarkan percepatan kendaraan yang lewat yakni dibawah atau diatas dari  $0\text{m/s}^2$ .

**ii. Berdasarkan hasil FFT, kira-kira range frequency berapa yang dihasilkan dari getaran akibat kendaraan yang lewat (untuk setiap jenis kendaraan akan menghasilkan frekuensi berbeda).**

Rata-rata frequency kendaraan yang lewat yakni sedang yakni  $>20\text{Hz}$ ~ dan kendaraan kecil  $<20\text{Hz}$ ~. Tidak ada kendaraan besar seperti truk yang lewat.

**d) Berdasarkan hasil FFT anda, telah didapatkan data di frekuensi berapa saja kendaraan yang lewat terdeteksi, misal kendaraan sedang dari 20 Hz sd. 40 Hz, kendaraan besar (truk dan bus) dari 30 Hz sd. 60 Hz. Dari hasil FFT tersebut, berikutnya adalah lakukan data atau signal filtering untuk membuang noise yang muncul dan tidak diinginkan pada sinyal accelerometer.**

**i. Filter apa yang anda gunakan (LPF, BPF, HPF, Notch ?). Jelaskan!**

Dalam eksperimen ini saya menggunakan High-Pass Filter dikarenakan untuk mengurangi noise kecil.

**ii. Lampirkan plot signal hasil filtering dan bandingkan dengan plot sinyal aslinya (sebelum dilakukan filtering).**

**Sebelum (abu) sesudah (hijau)**

