

Tugas UTS Pengantar Internet Of Things

Eksperiment

Eksperiment yang dilakukan adalah Menggunakan Aplikasi PhyPox Pada Android Type Advan GX Dengan Sample 100 Hz Dengan Kurun Waktu 180 Detik Dengan menghadapkan layer handphone ke atas , lokasi terdapat pada Jl.Kopo Depan Ayam Geprek Benu , dikarenakan jalan tersebut banyak kendaraan ringan seperti motor dan juga kendaraan sedang seperti mobil citycar SUV dan sebagainya maupun ada juga kendaraan berat yang telah keluar dari TOL Kopo untuk proyek pembangunan jalan flyover di jalan lewipanjang sampai caringin , ini adalah gambar untuk melakukan eksperiment yang dilakukan malam hari



Terdapat 2 Gambar Yg Pertama adalah gambar dimana handphone dihadapkan ke atas di sebelah Trotoar , dan juga gambar kedua adalah gambar dimana hasil sensing berupa screenshoot dari aplikasi android PhyPox yang menghasilkan nilai Linear Acceleration X,Y,Z

Jawaban No 1 Pada Soal

Pertama saya akan melakukan pengujian dan plotting data yaitu menggunakan Bahasa python dan melakukan import data excel dari hasil pengujian dan melakukan normalisasi data agar data mulai dari nilai 0 yaitu menggunakan rumus :

```
Az=Az-(sum(Az)/np.size(Az))
Ay=Ay-(sum(Ay)/np.size(Ay))
Ax=Ax-(sum(Ax)/np.size(Ax))
```

IMPORT DATA

```
In [3]: data_raw = pd.read_excel('C:/Users/stana/Documents/UTS S2/TUGAS UTS 2021-11-17 20-13-35.xls')
# data_raw.columns=["Time (s)", "Linear Acceleration x (m/s^2)", "Linear Acceleration y (m/s^2)", "Linear Acceleration z (m/s^2)"]
waktu = data_raw['Time (s)']
Ax = data_raw['Linear Acceleration x (m/s^2)']
Ay = data_raw['Linear Acceleration y (m/s^2)']
Az = data_raw['Linear Acceleration z (m/s^2)']
Az=Az-(sum(Az)/np.size(Az))
Ay=Ay-(sum(Ay)/np.size(Ay))
Ax=Ax-(sum(Ax)/np.size(Ax))
```

Lalu Melakukan Custom Warna Terhadap Hasil Plotting agar lebih jelas dan menarik untuk dilihat

Custom Warna

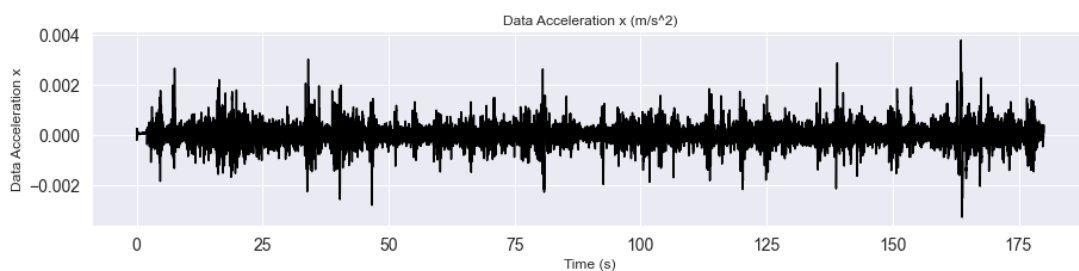
```
colors = ["windows red", "amber", "faded green", "dusty purple"]
sns.set(rc={"figure.figsize": (20,6), "axes.titlesize": 18, "axes.labelsize": 12,
           "xtick.labelsize": 14, "ytick.labelsize": 14 })
```

Ini Adalah Hasil dan Code program untuk linear acceleration sumbu X

ACCELERATION x

```
plt.subplots(figsize=(15,3))
plt.title('Data Acceleration x (m/s^2)', fontsize=12)
plt.plot(waktu,Ax,color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Data Acceleration x")
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

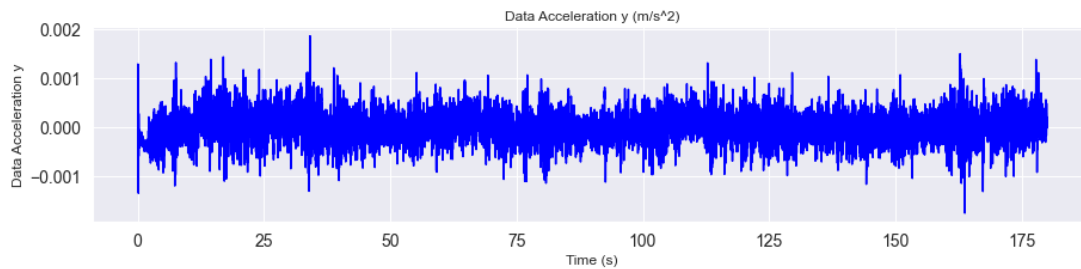


Ini Adalah Hasil dan Code program untuk linear acceleration sumbu Y

ACCELERATION y

```
plt.subplots(figsize=(15,3))
plt.title('Data Acceleration y (m/s^2)', fontsize=12)
plt.plot(waktu,Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Data Acceleration y")
plt.show
```

<function matplotlib.pyplot.show(close=None, block=None)>

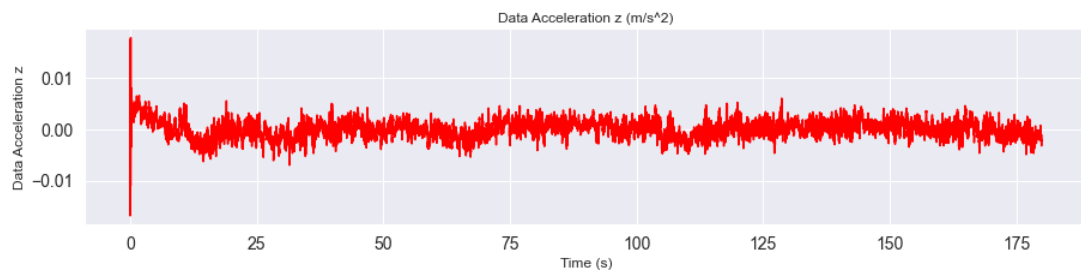


Ini Adalah Hasil dan Code program untuk linear acceleration sumbu Z

ACCELERATION z

```
plt.subplots(figsize=(15,3))
plt.title('Data Acceleration z (m/s^2)', fontsize=12)
plt.plot(waktu,Az,color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Data Acceleration z")
plt.show
```

<function matplotlib.pyplot.show(close=None, block=None)>



Penjelasan : Pada gambar atau hasil plotting yang digunakan menggunakan program Bahasa python adalah saya membuat subplots untuk gambar yang dihasilkan berukuran 15 x 3 dan memberikan judul Data Acceleration X,Y,Z , untuk acceleration X saya beri grafik berwarna hitam , untuk acceleraton Y saya beri grafik berwarna biru , lalu untuk acceleration z saya beri grafik berwarna merah dan saya memberi label Data sumbu x,y adalah waktu,data acceleration.

Hasil : Hasil yang ditunjukkan pada grafik X,Y,Z adalah yang lebih dominan adalah hasil dari sensing linear acceleration pada Sumbu X

Jawaban No 2 Pada Soal

Pada Soal Ke 2 ini saya akan melakukan FFT terhadap data accelerometer sumbu X,Y,Z dan berikut adalah code program Untuk melakukan frekuensi FFT terhadap acceleration sumbu X dan melakukan bandpass Filter terhadap hasil FFT Tersebut dan melakukan penglihatan pada range frekuensi yang terdapat , lalu melakukan pencarian Nilai Magnitude/Amplitude dan juga nilai Natural Frekuensi yang didapat , natural frekuensi adalah nilai Frekuensi Natural terhadap nilai magnitude yang terbesar nilainya

Code Program :

```
Tx = waktu[1]-waktu[0]
Nx = len(Ax)
yx = Ax.to_numpy()
yx = yx - np.ones_like(yx)*np.mean(yx) #Normalisasi

fft_yx = scipy.fftpack.fft(yx)
fft_xx = np.linspace(0.0, 1.0/(2.0*Tx), int(Nx/2))

plt.subplots(figsize=(15,3))
plt.plot(waktu,Ax,color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title("Acceleration x (m/s^2)")
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(fft_xx,2.0/Nx * np.abs(fft_yx[:Nx//2]),color = 'black')
plt.xlabel("Freq (Hz)")
plt.ylabel("Amplitude")
plt.title("Frequency Domain of Acceleration x ")
plt.show()

fft_xmax = 2.0/Nx * np.abs(fft_yx[:Nx//2])
magmax = max(fft_xmax)
# print(magmax)
for i in range(len(fft_xmax)):
    if fft_xmax[i]==magmax :
        nilaiFreq = fft_xx[i]
        nilaiMagnitude = fft_xmax[i]
        nilaiMaktu = waktu[i]
        print("Nilai Natural Frekuensi Sebelum Filter : ",fft_xx[i],'Hz')
        print("Nilai Magnitude/Amplitude Sebelum Filter : ",fft_xmax[i])

f_low_Ax = 0.1 #Cutoff bawah
f_high_Ax = 10 #Cutoff atas
fs_Ax = 1/Tx #Sampling Frequency

print("Range Frekuensi : ",f_low_Ax,"-",f_high_Ax,"Hz")

#Band pass Filter
# Filtered Data
filter_fftx = butter_bandpass_filter2(fft_yx,f_low_Ax, f_high_Ax, fs_Ax, order=4)

##response
b, a = butter_bandpass(f_low_Ax,f_high_Ax, fs_Ax, order=4)
# Plot the frequency response.
w, h = freqz(b, a, worN=64)

plt.subplots(figsize=(15,3))
plt.plot(fft_xx,2.0/Nx * np.abs(filter_fftx[:Nx//2]),color = 'black')
plt.axvline(f_low_Ax,color = 'green')
plt.axvline(f_high_Ax,color = 'green')
plt.xlabel("Freq (Hz)")
plt.ylabel("Amplitude Acceleration x")
plt.title("Frequency Domain of Filtered Acceleration x")
plt.show()

fft_xmax_filter = 2.0/Nx * np.abs(filter_fftx[:Nx//2])
magmax = max(fft_xmax_filter)
# print(magmax)
for i in range(len(fft_xmax_filter)):
    if fft_xmax_filter[i]==magmax :
        nilaiFreq = fft_xx[i]
        nilaiMagnitude = fft_xmax_filter[i]
        nilaiMaktu = waktu[i]
        print("Nilai Natural Frekuensi Sesudah Filter : ",fft_xx[i],'Hz')
        print("Nilai Magnitude/Amplitude Sesudah Filter : ",fft_xmax_filter[i])

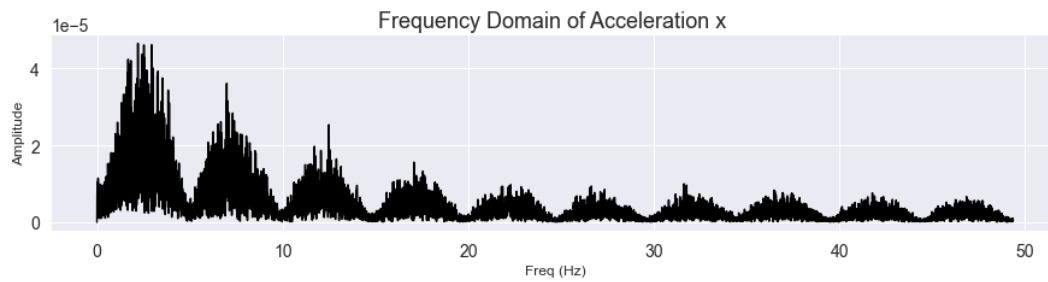
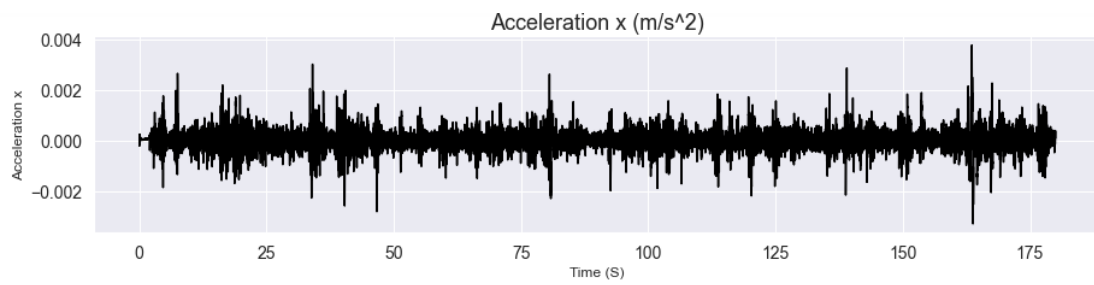
print("Range Frekuensi : ",f_low_Ax,"-",f_high_Ax,"Hz")
```

FUNGSI BANDPASS FILTER

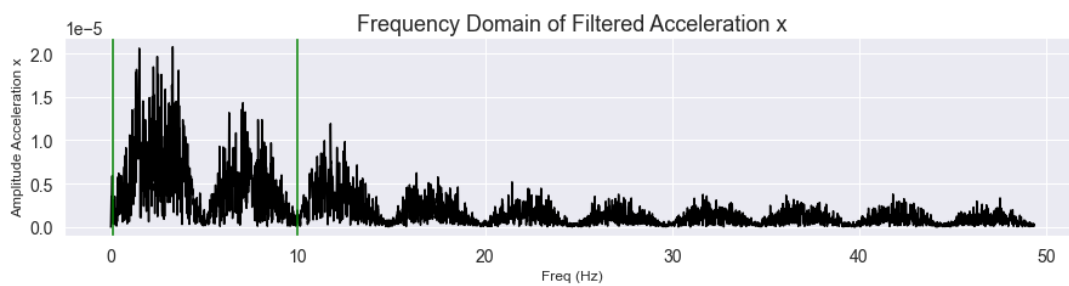
#Bandpass

```
def butter_bandpass(lowcut, highcut, fs, order):  
    nyq = 0.5 * fs  
    low = lowcut / nyq  
    high = highcut / nyq  
    b, a = butter(order, [low, high], btype='band')  
    return b, a  
  
def butter_bandpass_filter2(data, lowcut, highcut, fs, order):  
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)  
    y = lfilter(b, a, data)  
    # y =filtfilt(b, a, data)  
    return y
```

Ini Adalah Hasil Plotting data sebelum dan sesudah FFT dan juga hasil setelah bandpass Filter pada acceleration Sumbu X



Nilai Natural Frekuensi Sebelum Filter : 2.205989596264118 Hz
Nilai Magnitude/Amplitude Sebelum Filter : 4.6487493937480236e-05
Range Frekuensi : 0.1 - 10 Hz



Nilai Natural Frekuensi Sesudah Filter : 3.31731936768181 Hz
Nilai Magnitude/Amplitude Sesudah Filter : 2.0763661287672585e-05
Range Frekuensi : 0.1 - 10 Hz

Ini Adalah Hasil Plotting dan Code Program data sebelum dan sesudah FFT dan juga hasil setelah bandpass Filter pada acceleration Sumbu Y

Code Program :

```
Ty = waktu[1]-waktu[0]
Ny = len(Ay)
yy = Ay.to_numpy()
yy = yy - np.ones_like(yy)*np.mean(yy) #Normalisasi
fft_yy = scipy.fftpack.fft(yy)
fft_xy = np.linspace(0.0, 1.0/(2.0*Ty), int(Ny/2))

plt.subplots(figsize=(15,3))
plt.plot(waktu,Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Acceleration y (m/s^2)')
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(fft_xy,2.0/Ny * np.abs(fft_yy[:Ny//2]),color = 'blue')
plt.xlabel("Freq (Hz)")
plt.ylabel("Amplitude Acceleration y")
plt.title('Frequency Domain of Acceleration y ')
plt.show()

fft_ymax = 2.0/Ny * np.abs(fft_yy[:Ny//2])
magmax = max(fft_ymax)
# print(magmax)
for i in range(len(fft_ymax)):
    if fft_ymax[i]==magmax :
        nilaiFreq = fft_xy[i]
        nilaiMagnitude = fft_ymax[i]
        nilaiMaktu = waktu[i]
        print("Nilai Natural Frekuensi Sebelum Filter : ",fft_xy[i],'Hz')
        print("Nilai Magnitude/Amplitude Sebelum Filter : ",fft_ymax[i])

f_low_Ay = 0.1 #Cutoff bawah
f_high_Ay = 5 #Cutoff atas
fs_Ay = 1/Ty #Sampling Frequency

print("Range Frekuensi : ",f_low_Ay,"-",f_high_Ay,"Hz")

#Band pass filter
# Filtered Data
filter_ffty = butter_bandpass_filter2(fft_yy,f_low_Ay, f_high_Ay, fs_Ay, order=4)

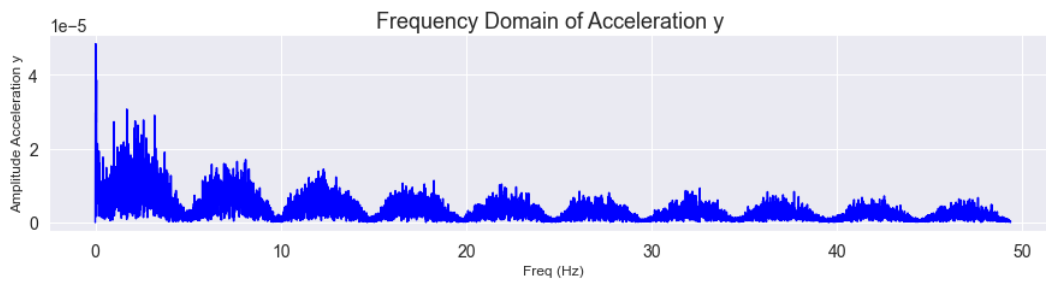
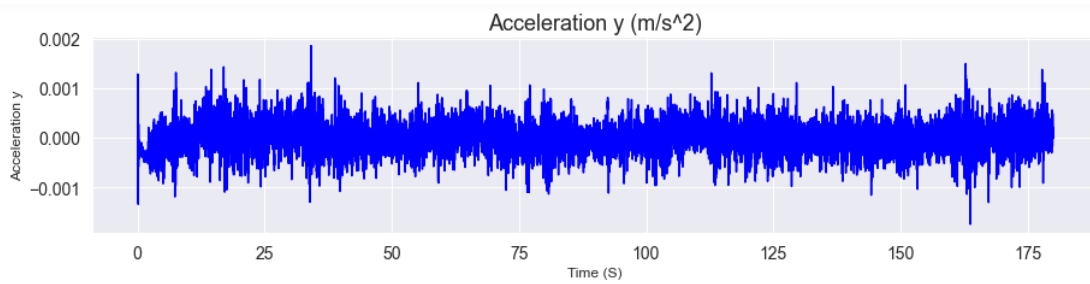
##response
b, a = butter_bandpass(f_low_Ay,f_high_Ay, fs_Ay, order=4)
# Plot the frequency response.
w, h = freqz(b, a, worN=64)

plt.subplots(figsize=(15,3))
plt.plot(fft_xy,2.0/Ny * np.abs(filter_ffty[:Ny//2]),color = 'blue')
plt.axvline(f_low_Ay,color = 'green')
plt.axvline(f_high_Ay,color = 'green')
plt.xlabel("Freq (Hz)")
plt.ylabel("Amplitude Acceleration y")
plt.title('Frequency Domain of Filtered Acceleration y')
plt.show()

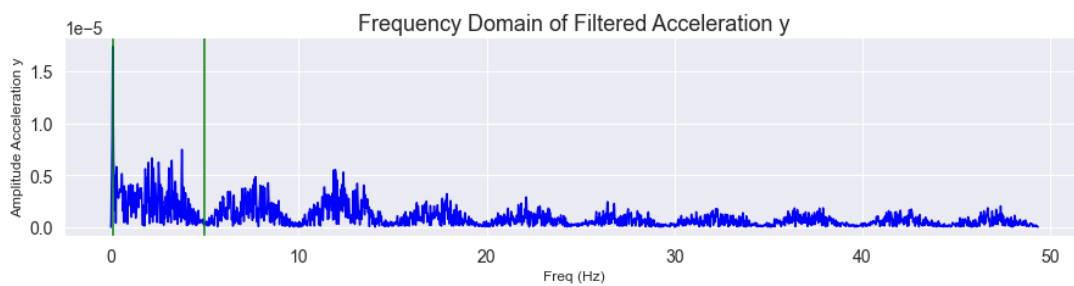
fft_ymax_filter = 2.0/Ny * np.abs(filter_ffty[:Ny//2])
magmax = max(fft_ymax_filter)
# print(magmax)
for i in range(len(fft_ymax_filter)):
    if fft_ymax_filter[i]==magmax :
        nilaiFreq = fft_xy[i]
        nilaiMagnitude = fft_ymax_filter[i]
        nilaiMaktu = waktu[i]
        print("Nilai Natural Frekuensi Sesudah Filter : ",fft_xy[i],'Hz')
        print("Nilai Magnitude/Amplitude Sesudah Filter : ",fft_ymax_filter[i])

print("Range Frekuensi : ",f_low_Ay,"-",f_high_Ay,"Hz")
```

Hasil Plotting data sebelum dan sesudah FFT dan setelah Bandpass Filter



Nilai Natural Frekuensi Sebelum Filter : 0.022226595428353833 Hz
Nilai Magnitude/Amplitude Sebelum Filter : 4.839065435297858e-05
Range Frekuensi : 0.1 - 5 Hz



Nilai Natural Frekuensi Sesudah Filter : 0.1055763282846807 Hz
Nilai Magnitude/Amplitude Sesudah Filter : 1.7374504357599193e-05
Range Frekuensi : 0.1 - 5 Hz

Ini Adalah Hasil Plotting dan Code Program data sebelum dan sesudah FFT dan juga hasil setelah bandpass Filter pada acceleration Sumbu

Code Program :

```
T2 = waktu[1]-waktu[0]
Nz = len(Az)
y2 = Az.to_numpy()
y2 = y2 - np.ones_like(y2)*np.mean(y2) #Normalisasi
fft_y2 = scipy.fftpack.fft(y2)
fft_x2 = np.linspace(0.0, 1.0/(2.0*T2), int(Nz/2))

plt.subplots(figsize=(15,3))
plt.plot(fft_x2,2.0/Nz * np.abs(fft_y2[:Nz//2]),color = 'red')
plt.xlabel("Freq (Hz)")
plt.ylabel("Amplitude Acceleration z")
plt.title("Acceleration z (m/s^2)")
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(fft_x2,2.0/Nz * np.abs(fft_y2[:Nz//2]),color = 'red')
plt.xlabel("Freq (Hz)")
plt.ylabel("Amplitude Acceleration z")
plt.title("Frequency Domain of Acceleration z ")
plt.show()

fft_zmax = 2.0/Nz * np.abs(fft_y2[:Nz//2])
magmax = max(fft_zmax)
# print(magmax)
for i in range(len(fft_zmax)):
    if fft_zmax[i]==magmax :
        nilaiFreq = fft_x2[i]
        nilaiMagnitude = fft_zmax[i]
        nilaiMaktu = waktu[i]
        print("Nilai Natural Frekuensi Sebelum Filter : ",fft_x2[i],'Hz')
        print("Nilai Magnitude/Amplitude Sebelum Filter : ",fft_zmax[i])

F_low_Az = 0.1 #Cutoff bawah
F_high_Az = 3 #Cutoff atas
fs_Az = 1/T2 #Sampling Frequency

print("Range Frekuensi : ",F_low_Az,"-",F_high_Az,"Hz")

#Band pass filter
# Filtered Data
filter_fft2 = butter_bandpass_filter2(fft_y2,F_low_Az, F_high_Az, fs_Az, order=4)

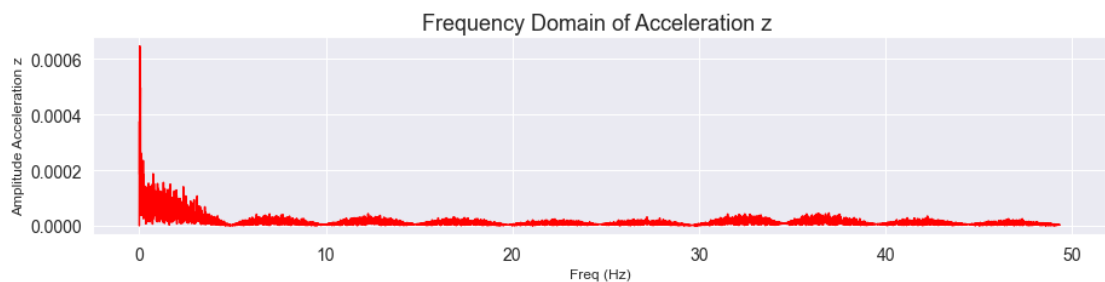
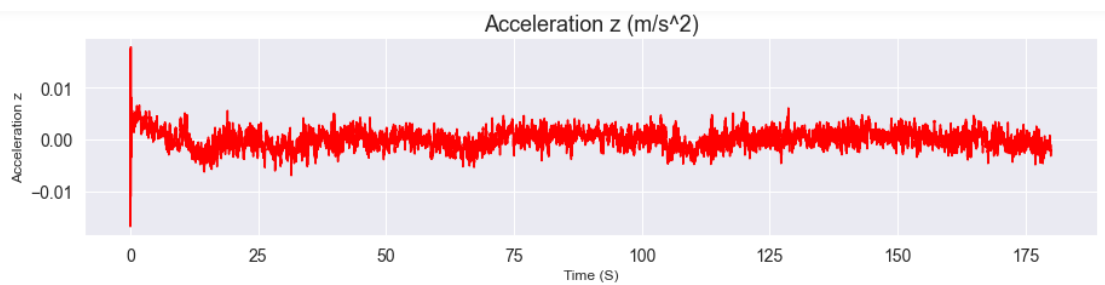
##response
b, a = butter_bandpass(F_low_Az,F_high_Az, fs_Az, order=4)
# Plot the frequency response.
w, h = freqz(b, a, worN=64)

plt.subplots(figsize=(15,3))
plt.plot(fft_x2,2.0/Nz * np.abs(filter_fft2[:Nz//2]),color = 'red')
plt.axvline(F_low_Az,color = 'green')
plt.axvline(F_high_Az,color = 'green')
plt.xlabel("Freq (Hz)")
plt.ylabel("Amplitude Acceleration z")
plt.title("Frequency Domain of Filtered Acceleration z")
plt.show()

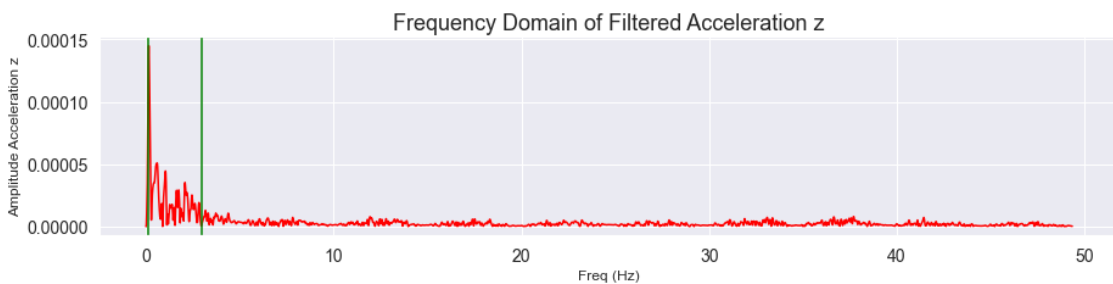
fft_zmax_filter = 2.0/Nz * np.abs(filter_fft2[:Nz//2])
magmax = max(fft_zmax_filter)
# print(magmax)
for i in range(len(fft_zmax_filter)):
    if fft_zmax_filter[i]==magmax :
        nilaiFreq = fft_x2[i]
        nilaiMagnitude = fft_zmax_filter[i]
        nilaiMaktu = waktu[i]
        print("Nilai Natural Frekuensi Sesudah Filter : ",fft_x2[i],'Hz')
        print("Nilai Magnitude/Amplitude Sesudah Filter : ",fft_zmax_filter[i])

print("Range Frekuensi : ",F_low_Az,"-",F_high_Az,"Hz")
```

Hasil Plotting data sebelum dan sesudah FFT dan setelah Bandpass Filter



Nilai Natural Frekuensi Sebelum Filter : 0.022226595428353833 Hz
Nilai Magnitude/Amplitude Sebelum Filter : 0.0006468952181878609
Range Frekuensi : 0.1 - 3 Hz



Nilai Natural Frekuensi Sesudah Filter : 0.15002951914138837 Hz
Nilai Magnitude/Amplitude Sesudah Filter : 0.0001448949314022299
Range Frekuensi : 0.1 - 3 Hz

Penjelasan : Saat Melakukan FFT terhadap Linear Acceleration X,Y,Z terdapat perbedaan dimana nilai beberapa frekuensi yang dihasilkan maupun nilai magnitude/amplitude : ini adalah nilai Frekuensi, Magnitude/Amplitude dan juga range frekuensi yang didapat pada Linear Acceleration X,Y,Z :

1. Linear Acceleration X

Sebelum Filtering Bandpass

Nilai Natural Frekuensi Sebelum Filter : 2.205989596264118 Hz
Nilai Magnitude/Amplitude Sebelum Filter : 4.6487493937480236e-05
Range Frekuensi : 0.1 - 10 Hz

Sesudah Filtering Bandpass

Nilai Natural Frekuensi Sesudah Filter : 3.31731936768181 Hz
Nilai Magnitude/Amplitude Sesudah Filter : 2.0763661287672585e-05
Range Frekuensi : 0.1 - 10 Hz

2. Linear Acceleration Y

Sebelum Filtering Bandpass

Nilai Natural Frekuensi Sebelum Filter : 0.022226595428353833 Hz
Nilai Magnitude/Amplitude Sebelum Filter : 4.839065435297858e-05
Range Frekuensi : 0.1 - 5 Hz

Sesudah Filtering Bandpass

Nilai Natural Frekuensi Sesudah Filter : 0.1055763282846807 Hz
Nilai Magnitude/Amplitude Sesudah Filter : 1.7374504357599193e-05
Range Frekuensi : 0.1 - 5 Hz

3. Linear Acceleration Z

Sebelum Filtering Bandpass

Nilai Natural Frekuensi Sebelum Filter : 0.022226595428353833 Hz
Nilai Magnitude/Amplitude Sebelum Filter : 0.0006468952181878609
Range Frekuensi : 0.1 - 3 Hz

Sesudah Filtering Bandpass

Nilai Natural Frekuensi Sesudah Filter : 0.15002951914138837 Hz
Nilai Magnitude/Amplitude Sesudah Filter : 0.0001448949314022299
Range Frekuensi : 0.1 - 3 Hz

Hasil : Dapat Dilihat dari beberapa Nilai dan Range Frekuensi Yang Didapat adalah Data Linear Acceleration Sumbu X lebih signifikan untuk menemukan nilai frekuensi pada eksperimen yang dilakukan yaitu mendapatkan nilai frekuensi adalah 2 Hz dan 3 Hz Setelah Filtering dengan beberapa hasil dibandingkan dengan yang lain yaitu sumbu Linear Acceleration Y & Z menghasilkan nilai frekuensi yang jauh berbeda. Karna memungkinkan pada device menghadap ke atas adalah Gerakan getaran yang berlangsung adalah pada linear accelerometer Sumbu x. maka nilai nilai dari linear acceleration sumbu x dapat bervariasi dan berubah2 dan dapat dilihat pada grafik.

Jawaban No 3 Pada Soal

Pada Soal ini saya memiliki paham untuk konsep mendapatkan jenis kendaraan yaitu dengan menggunakan periode dari waktu saya membuat 6 periode waktu yaitu dari waktu 0-30 detik (Periode 1) lalu 30-60 detik (Periode 2) lalu 60-90 detik (Periode 3) lalu 90-120 detik (Periode 4) 120-150 detik (Periode 5) dan 150-180 detik (Periode 6). Yang Dimana Saya akan menghitung nilai rata2 terhadap data periode 1-5 masing2 data pada acceleration X,Y,Z yang sudah difilter menggunakan bandpass filter lalu saya akan membuat prediksi data dari data rata-rata tersebut dengan urutan ascending bila data paling kecil akan diasumsikan dengan data “Sepeda Motor 1 Orang” lalu data terkecil ke 2 adalah “Sepeda Motor 2 Orang” lalu data ke 3 yaitu “Mobil 1 Orang” , lalu data ke 4 yaitu “Mobil 4-6 Orang” Lalu data Ke 5 adalah “Truck Kosong” , dan data ke 6 atau terbesar rata-rata dari grafik yaitu “Truck Full Kapasitas” lalu akan saya rata2kan Kembali terhadap data tsb dengan accelerometer Sumbu X,Y,Z. berikut hasil penelitian saya terhadap eksperiment

Code Program Linear Acceleration X:

```
f_low_Ax = 0.1 #Cutoff bawah
f_high_Ax = 10 #Cutoff atas
fs_Ax = 1/Tx #Sampling Frequency

#Band pass filter
# Filtered Data
Filter_Ax = butter_bandpass_filter2(Ax, f_low_Ax, f_high_Ax, fs_Ax, order=4)

##response
b, a = butter_bandpass(f_low_Ax, f_high_Ax, fs_Ax, order=4)
# Plot the frequency response.
w, h = freqz(b, a, worN=64)

plt.subplots(figsize=(15,3))
plt.plot(waktu, Ax, color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title('Acceleration x (m/s^2)')
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, Filter_Ax, color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title('Linear Acceleration x (m/s^2) of Filtered Bandpass')
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, Filter_Ax, color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title('Linear Acceleration x (m/s^2) of Filtered Bandpass Periode Waktu 0-30 second')
plt.xlim([0, 30])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, Filter_Ax, color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title('Linear Acceleration x (m/s^2) of Filtered Bandpass Periode Waktu 30-60 second')
plt.xlim([30, 60])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, Filter_Ax, color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title('Linear Acceleration x (m/s^2) of Filtered Bandpass Periode Waktu 60-90 second')
plt.xlim([60, 90])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, Filter_Ax, color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title('Linear Acceleration x (m/s^2) of Filtered Bandpass Periode Waktu 90-120 second')
plt.xlim([90, 120])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, Filter_Ax, color = 'black')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration x")
plt.title('Linear Acceleration x (m/s^2) of Filtered Bandpass Periode Waktu 120-150 second')
plt.xlim([120, 150])
plt.show()
```

```

print("***** Linear Acceleration x Classification *****")
sumx1 = 0
ratax1 = 0
for x in range(0,3000):
    if(Filter_Ax[x]>=0):
        sumx1 = sumx1 + filter_Ax[x]

ratax1 = sumx1/3000
print ("Rata-Rata Periode Ke 1 Yaitu Detik 0-30 : ",ratax1)

sumx2 = 0
ratax2 = 0
for x in range(3000,6000):
    if(Filter_Ax[x]>=0):
        sumx2 = sumx2 + filter_Ax[x]

ratax2 = sumx2/3000
print ("Rata-Rata Periode Ke 2 Yaitu Detik 30-60 : ",ratax2)

sumx3 = 0
ratax3 = 0
for x in range(6000,9000):
    if(Filter_Ax[x]>=0):
        sumx3 = sumx3 + filter_Ax[x]

ratax3 = sumx3/3000
print ("Rata-Rata Periode Ke 3 Yaitu Detik 60-90 : ",ratax3)

sumx4 = 0
ratax4 = 0
for x in range(9000,12000):
    if(Filter_Ax[x]>=0):
        sumx4 = sumx4 + filter_Ax[x]

ratax4 = sumx4/3000
print ("Rata-Rata Periode Ke 4 Yaitu Detik 90-120 : ",ratax4)

sumx5 = 0
ratax5 = 0
for x in range(12000,15000):
    if(Filter_Ax[x]>=0):
        sumx5 = sumx5 + filter_Ax[x]

ratax5 = sumx5/3000
print ("Rata-Rata Periode Ke 5 Yaitu Detik 120-150 : ",ratax5)

sumx6 = 0
ratax6 = 0
for x in range(15000,17756):
    if(Filter_Ax[x]>=0):
        sumx6 = sumx6 + filter_Ax[x]

ratax6 = sumx6/(17756-15000)
print ("Rata-Rata Periode Ke 6 Yaitu Detik 150-180 : ",ratax6)

data_Rata_Rata_Ax = [ratax1,ratax2,ratax3,ratax4,ratax5,ratax6]
ranking_Ax = sorted(data_Rata_Rata_Ax)
data_klasifikasi = ['Sepeda Motor 1 Orang','Sepeda Motor 2 Orang','Mobil 1 Orang','Mobil 4-6 Orang','Truck Kosong','Truck Full K

for x in range(0,len(ranking_Ax)):
    if(ranking_Ax[x]==ratax1):
        print("Periode Ke - 1 = 0-30 Detik")
        print("Rata-Rata = ",ranking_Ax[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ax)):
    if(ranking_Ax[x]==ratax2):
        print("Periode Ke - 2 = 30-60 Detik")
        print("Rata-Rata = ",ranking_Ax[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ax)):
    if(ranking_Ax[x]==ratax3):
        print("Periode Ke - 3 = 60-90 Detik")
        print("Rata-Rata = ",ranking_Ax[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ax)):
    if(ranking_Ax[x]==ratax4):
        print("Periode Ke - 4 = 90-120 Detik")
        print("Rata-Rata = ",ranking_Ax[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ax)):
    if(ranking_Ax[x]==ratax5):
        print("Periode Ke - 5 = 120-150 Detik")
        print("Rata-Rata = ",ranking_Ax[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

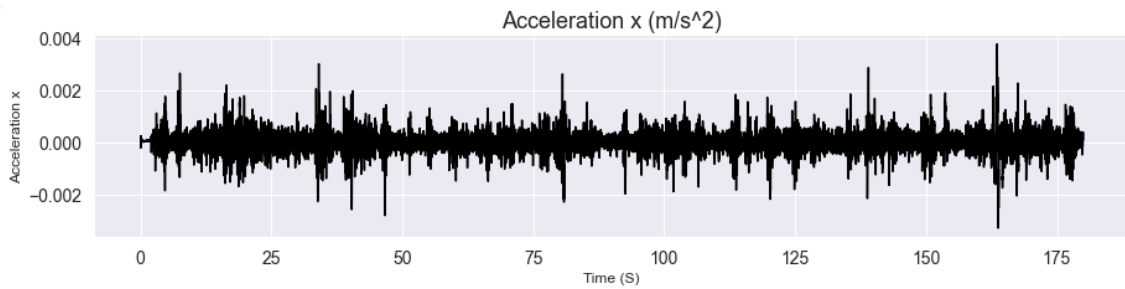
print("")

for x in range(0,len(ranking_Ax)):
    if(ranking_Ax[x]==ratax6):
        print("Periode Ke - 6 = 150-180 Detik")
        print("Rata-Rata = ",ranking_Ax[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

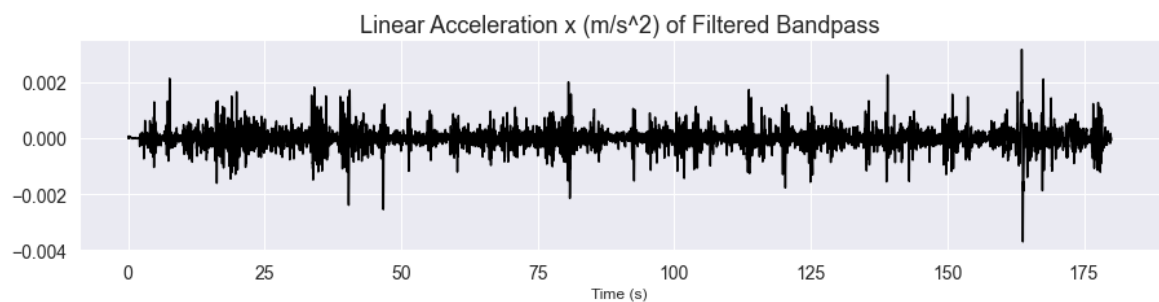
```

Berikut adalah Hasil eksekusi program Linear Acceleration X :

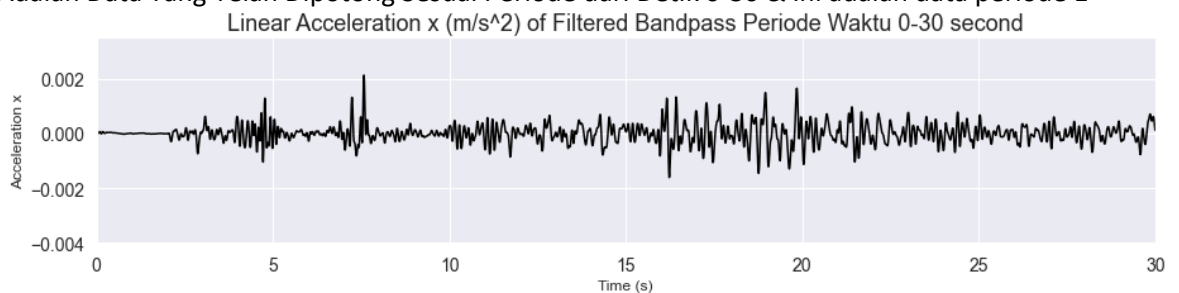
1. Adalah Data Awal dari Linear Acceleration Sumbu X



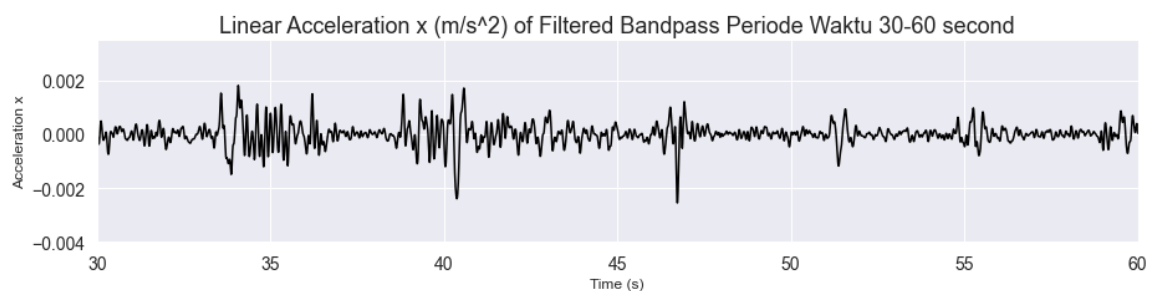
2. Adalah Data Yg Telah Di Filter Menggunakan Bandpass



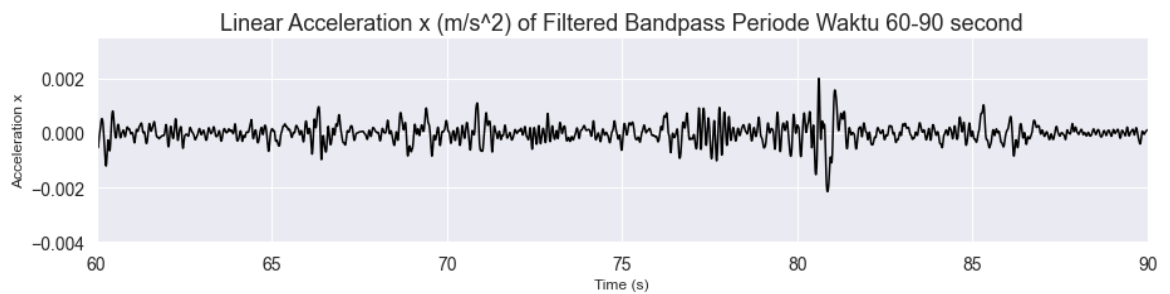
3. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 0-30 & Ini adalah data periode 1



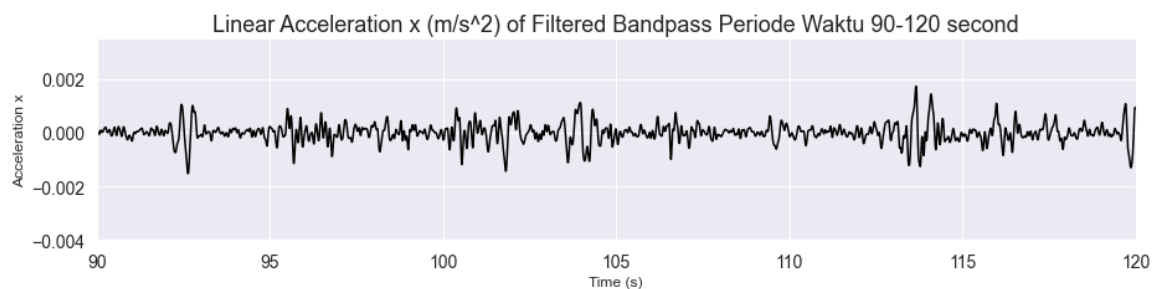
4. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 30-60 & Ini adalah data periode 2



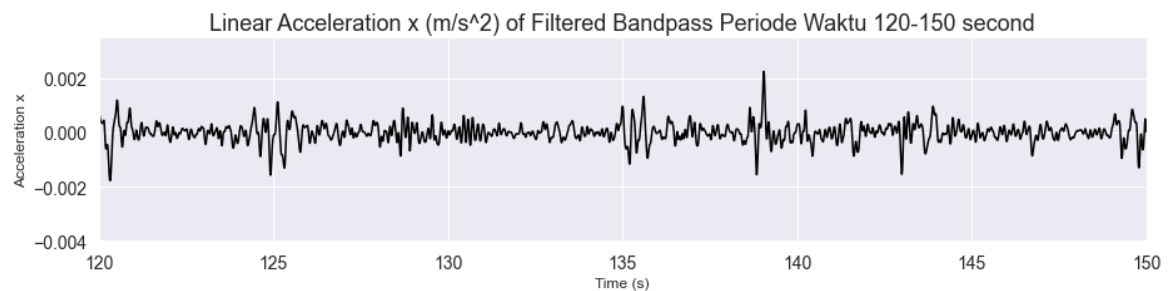
5. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 60-90 & Ini adalah data periode 3



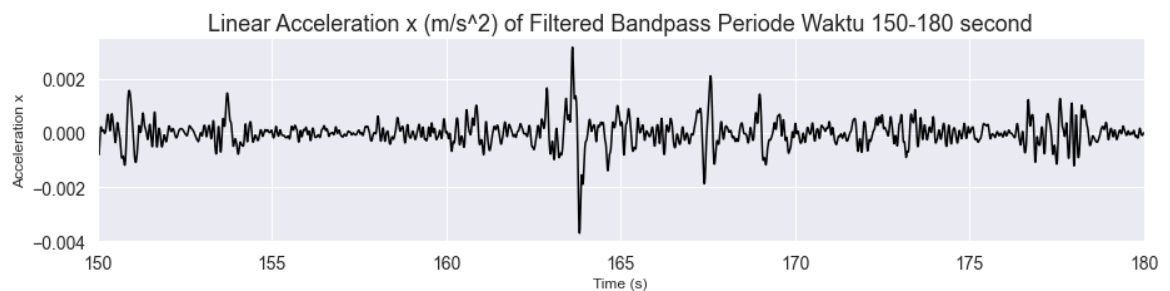
6. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 90-120 & Ini adalah data periode 4



7. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 120-150 & Ini adalah data periode 5



8. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 150-180 & Ini adalah data periode 6



9. Dan Ini Adalah Hasil Rata Rata Dari Periode ke 1 hingga period ke 6

```
===== Linear Acceleration x Clasification =====
Rata-Rata Periode Ke 1 Yaitu Detik 0-30      : 0.0001228585552309629
Rata-Rata Periode Ke 2 Yaitu Detik 30-60     : 0.00013105428576485345
Rata-Rata Periode Ke 3 Yaitu Detik 60-90     : 0.00011396477449184393
Rata-Rata Periode Ke 4 Yaitu Detik 90-120    : 0.00012616470121481812
Rata-Rata Periode Ke 5 Yaitu Detik 120-150   : 0.00012508102531442967
Rata-Rata Periode Ke 6 Yaitu Detik 150-180   : 0.0001536609280330769
```

10. Lalu Data Tsb Akan di ranking dengan menggunakan sorting secara ascending lalu dari data ascending tsb dicocokkan dengan data klasifikasi kendaraan dan menghasilkan berupa urutan data periode .

```
Periode Ke - 1      = 0-30 Detik
Rata-Rata          = 0.0001228585552309629
Prediksi Kendaraan = Sepeda Motor 2 Orang

Periode Ke - 2      = 30-60 Detik
Rata-Rata          = 0.00013105428576485345
Prediksi Kendaraan = Truck Kosong

Periode Ke - 3      = 60-90 Detik
Rata-Rata          = 0.00011396477449184393
Prediksi Kendaraan = Sepeda Motor 1 Orang

Periode Ke - 4      = 90-120 Detik
Rata-Rata          = 0.00012616470121481812
Prediksi Kendaraan = Mobil 4-6 Orang

Periode Ke - 5      = 120-150 Detik
Rata-Rata          = 0.00012508102531442967
Prediksi Kendaraan = Mobil 1 Orang

Periode Ke - 6      = 150-180 Detik
Rata-Rata          = 0.0001536609280330769
Prediksi Kendaraan = Truck Full Kapasitas
```

Gambar Diatas Adalah Menunjukkan hasil klasifikasi dengan ide yang saya jelaskan diatas yang diambil dari data Linear Acceleration Sumbu X

Code Program Linear Acceleration Y:

```
f_low_Ay = 0.1 #Cutoff bawah
f_high_Ay = 5 #Cutoff atas
fs_Ay = 1/1y #Sampling Frequency

#Band pass filter
# Filtered Data
filter_Ay = butter_bandpass_filter2(Ay, f_low_Ay, f_high_Ay, fs_Ay, order=4)

##response
b, a = butter_bandpass(f_low_Ay, f_high_Ay, fs_Ay, order=4)
# Plot the frequency response.
w, h = freqz(b, a, worN=64)

plt.subplots(figsize=(15,3))
plt.plot(waktu,Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Acceleration y (m/s^2)')
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu,filter_Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Linear Acceleration y (m/s^2) of Filtered Bandpass')
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu,filter_Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Linear Acceleration y (m/s^2) of Filtered Bandpass Periode Waktu 0-30 second')
plt.xlim([0, 30])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu,filter_Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Linear Acceleration y (m/s^2) of Filtered Bandpass Periode Waktu 30-60 second')
plt.xlim([30, 60])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu,filter_Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Linear Acceleration y (m/s^2) of Filtered Bandpass Periode Waktu 60-90 second')
plt.xlim([60, 90])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu,filter_Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Linear Acceleration y (m/s^2) of Filtered Bandpass Periode Waktu 90-120 second')
plt.xlim([90, 120])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu,filter_Ay,color = 'blue')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Linear Acceleration y (m/s^2) of Filtered Bandpass Periode Waktu 120-150 second')
plt.xlim([120, 150])
plt.show()
```

```

print("===== Linear Acceleration y Clasification =====")
sumy1 = 0
ratay1 = 0
for x in range(0,3000):
    if(filter_Ay[x]>=0):
        sumy1 = sumy1 + filter_Ay[x]

ratay1 = sumy1/3000
print ("Rata-Rata Periode Ke 1 Valtu Detik 0-30 : ",ratay1)

sumy2 = 0
ratay2 = 0
for x in range(3000,6000):
    if(filter_Ay[x]>=0):
        sumy2 = sumy2 + filter_Ay[x]

ratay2 = sumy2/3000
print ("Rata-Rata Periode Ke 2 Valtu Detik 30-60 : ",ratay2)

sumy3 = 0
ratay3 = 0
for x in range(6000,9000):
    if(filter_Ay[x]>=0):
        sumy3 = sumy3 + filter_Ay[x]

ratay3 = sumy3/3000
print ("Rata-Rata Periode Ke 3 Valtu Detik 60-90 : ",ratay3)

sumy4 = 0
ratay4 = 0
for x in range(9000,12000):
    if(filter_Ay[x]>=0):
        sumy4 = sumy4 + filter_Ay[x]

ratay4 = sumy4/3000
print ("Rata-Rata Periode Ke 4 Valtu Detik 90-120 : ",ratay4)

sumy5 = 0
ratay5 = 0
for x in range(12000,15000):
    if(filter_Ay[x]>=0):
        sumy5 = sumy5 + filter_Ay[x]

ratay5 = sumy5/3000
print ("Rata-Rata Periode Ke 5 Valtu Detik 120-150 : ",ratay5)

sumy6 = 0
ratay6 = 0
for x in range(15000,17756):
    if(filter_Ay[x]>=0):
        sumy6 = sumy6 + filter_Ay[x]

ratay6 = sumy6/(17756-15000)
print ("Rata-Rata Periode Ke 6 Valtu Detik 150-180 : ",ratay6)

data_Rata_Rata_Ay = [ratay1,ratay2,ratay3,ratay4,ratay5,ratay6]
ranking_Ay = sorted(data_Rata_Rata_Ay)
data_klasifikasi = ['Sepeda Motor 1 Orang','Sepeda Motor 2 Orang','Mobil 1 Orang','Mobil 4-6 Orang','Truck Kosong','Truck Full K

for x in range(0,len(ranking_Ay)):
    if(ranking_Ay[x]==ratay1):
        print("Periode Ke - 1 = 0-30 Detik")
        print("Rata-Rata = ",ranking_Ay[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ay)):
    if(ranking_Ay[x]==ratay2):
        print("Periode Ke - 2 = 30-60 Detik")
        print("Rata-Rata = ",ranking_Ay[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ay)):
    if(ranking_Ay[x]==ratay3):
        print("Periode Ke - 3 = 60-90 Detik")
        print("Rata-Rata = ",ranking_Ay[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ay)):
    if(ranking_Ay[x]==ratay4):
        print("Periode Ke - 4 = 90-120 Detik")
        print("Rata-Rata = ",ranking_Ay[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_Ay)):
    if(ranking_Ay[x]==ratay5):
        print("Periode Ke - 5 = 120-150 Detik")
        print("Rata-Rata = ",ranking_Ay[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

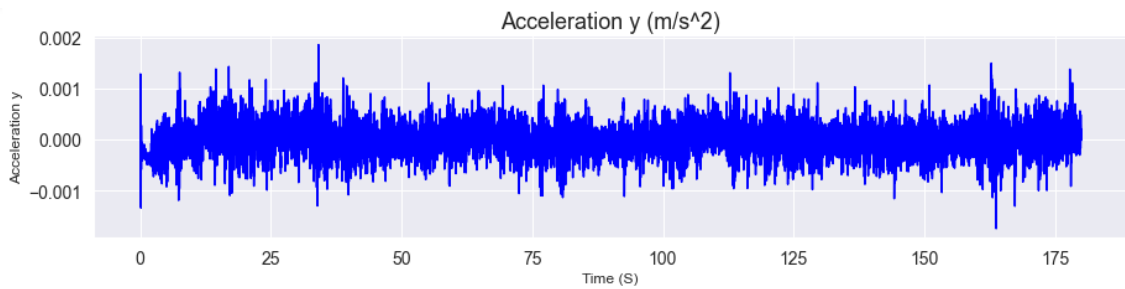
print("")

for x in range(0,len(ranking_Ay)):
    if(ranking_Ay[x]==ratay6):
        print("Periode Ke - 6 = 150-180 Detik")
        print("Rata-Rata = ",ranking_Ay[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

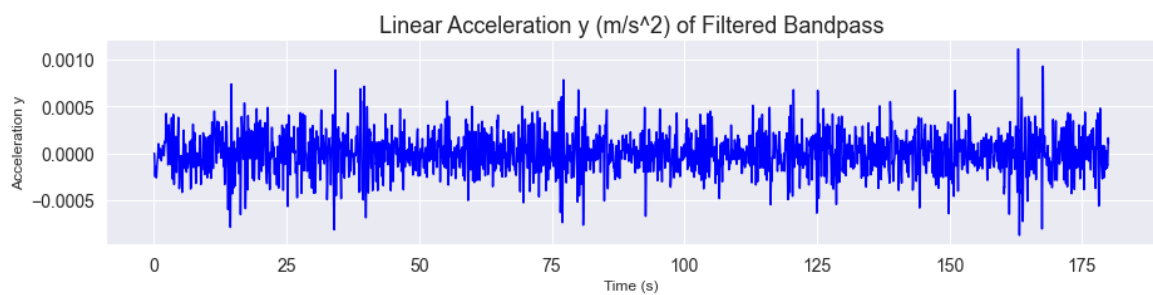
```

Berikut adalah Hasil eksekusi program Linear Acceleration y :

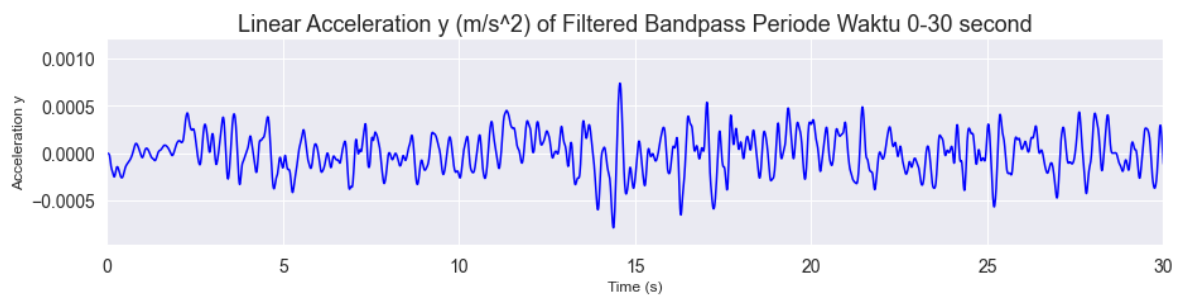
1. Adalah Data Awal dari Linear Acceleration Sumbu Y



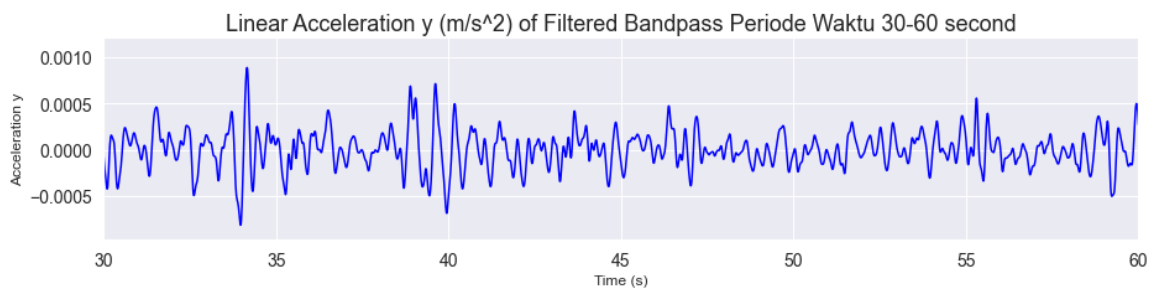
2. Adalah Data Yg Telah Di Filter Menggunakan Bandpass



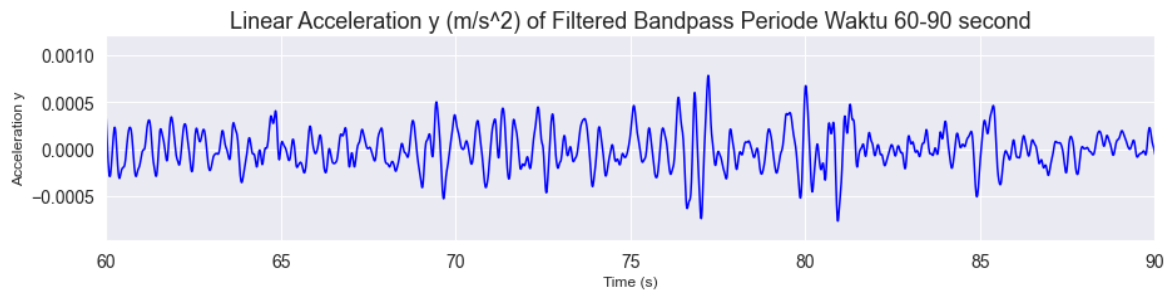
3. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 0-30 & Ini adalah data periode 1



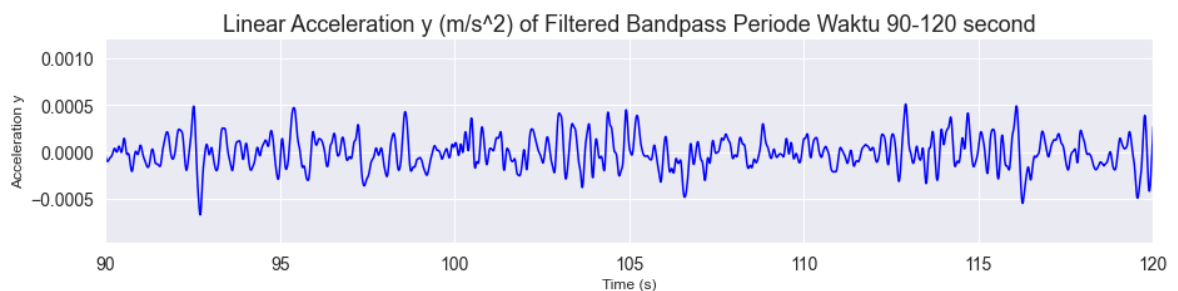
4. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 30-60 & Ini adalah data periode 2



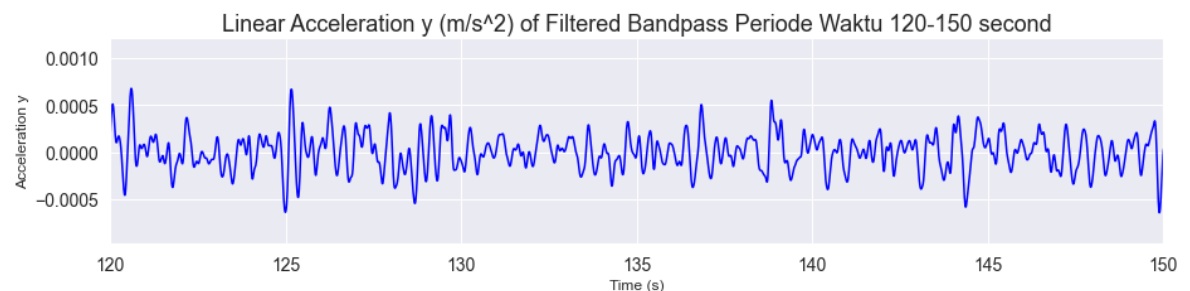
5. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 60-90 & Ini adalah data periode 3



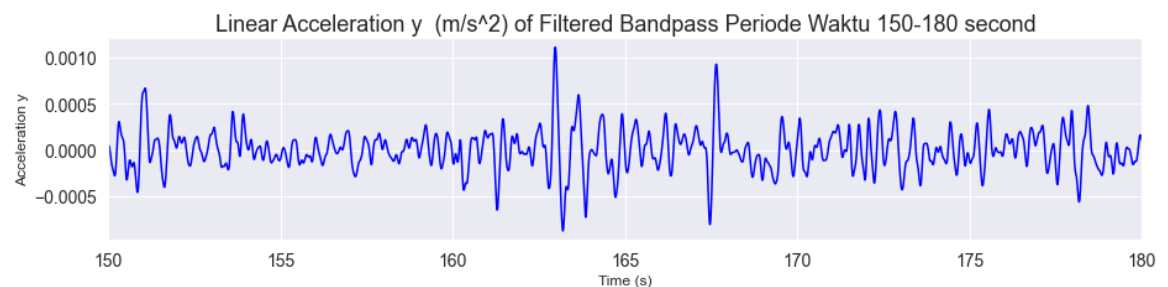
6. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 90-120 & Ini adalah data periode 4



7. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 120-150 & Ini adalah data periode 5



8. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 150-180 & Ini adalah data periode 6



9. Dan Ini Adalah Hasil Rata Rata Dari Periode ke 1 hingga period ke 6

```
===== Linear Acceleration y Clasification =====
Rata-Rata Periode Ke 1 Yaitu Detik 0-30      : 8.128113458674513e-05
Rata-Rata Periode Ke 2 Yaitu Detik 30-60     : 7.583868310076104e-05
Rata-Rata Periode Ke 3 Yaitu Detik 60-90     : 7.59808756217604e-05
Rata-Rata Periode Ke 4 Yaitu Detik 90-120    : 6.847801409837671e-05
Rata-Rata Periode Ke 5 Yaitu Detik 120-150   : 7.515379284917109e-05
Rata-Rata Periode Ke 6 Yaitu Detik 150-180   : 7.695744139444911e-05
```

10. Lalu Data Tsb Akan di ranking dengan menggunakan sorting secara ascending lalu dari data ascending tsb dicocokkan dengan data klasifikasi kendaraan dan menghasilkan berupa urutan data periode .

Periode Ke - 1 = 0-30 Detik
Rata-Rata = 8.128113458674513e-05
Prediksi Kendaraan = Truck Full Kapasitas

Periode Ke - 2 = 30-60 Detik
Rata-Rata = 7.583868310076104e-05
Prediksi Kendaraan = Mobil 1 Orang

Periode Ke - 3 = 60-90 Detik
Rata-Rata = 7.59808756217604e-05
Prediksi Kendaraan = Mobil 4-6 Orang

Periode Ke - 4 = 90-120 Detik
Rata-Rata = 6.847801409837671e-05
Prediksi Kendaraan = Sepeda Motor 1 Orang

Periode Ke - 5 = 120-150 Detik
Rata-Rata = 7.515379284917109e-05
Prediksi Kendaraan = Sepeda Motor 2 Orang

Periode Ke - 6 = 150-180 Detik
Rata-Rata = 7.695744139444911e-05
Prediksi Kendaraan = Truck Kosong

Gambar Diatas Adalah Menunjukkan hasil klasifikasi dengan ide yang saya jelaskan diatas yang diambil dari data Linear Acceleration Sumbu Y

Code Program Linear Acceleration Z:

```
f_low_Az = 0.1 #Cutoff bawah
f_high_Az = 5 #Cutoff atas
fs_Az = 1/Tz #Sampling Frequency

#Band pass Filter
# Filtered Data
filter_Az = butter_bandpass_filter2(Az, f_low_Az, f_high_Az, fs_Az, order=4)

##response
b, a = butter_bandpass(f_low_Az, f_high_Az, fs_Az, order=4)
# Plot the frequency response.
w, h = freqz(b, a, worN=64)

plt.subplots(figsize=(15,3))
plt.plot(waktu, Az, color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration z")
plt.title('Acceleration z (m/s^2)')
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, filter_Az, color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration z")
plt.title('Linear Acceleration z (m/s^2) of Filtered Bandpass')
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, filter_Az, color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration y")
plt.title('Linear Acceleration y (m/s^2) of Filtered Bandpass Periode Waktu 0-30 second')
plt.xlim([0, 30])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, filter_Az, color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration z")
plt.title('Linear Acceleration z (m/s^2) of Filtered Bandpass Periode Waktu 30-60 second')
plt.xlim([30, 60])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, filter_Az, color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration z")
plt.title('Linear Acceleration z (m/s^2) of Filtered Bandpass Periode Waktu 60-90 second')
plt.xlim([60, 90])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, filter_Az, color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration z")
plt.title('Linear Acceleration z (m/s^2) of Filtered Bandpass Periode Waktu 90-120 second')
plt.xlim([90, 120])
plt.show()

plt.subplots(figsize=(15,3))
plt.plot(waktu, filter_Az, color = 'red')
plt.xlabel("Time (s)")
plt.ylabel("Acceleration z")
plt.title('Linear Acceleration z (m/s^2) of Filtered Bandpass Periode Waktu 120-150 second')
plt.xlim([120, 150])
plt.show()
```

```

print("===== Linear Acceleration 2 Clasification =====")
sum21 = 0
rata21 = 0
for x in range(0,3000):
    if(filter_A2[x]>=0):
        sum21 = sum21 + filter_A2[x]

rata21 = sum21/3000
print ("Rata-Rata Periode Ke 1 Yaitu Detik 0-30 : ",rata21)

sum22 = 0
rata22 = 0
for x in range(3000,6000):
    if(filter_A2[x]>=0):
        sum22 = sum22 + filter_A2[x]

rata22 = sum22/3000
print ("Rata-Rata Periode Ke 2 Yaitu Detik 30-60 : ",rata22)

sum23 = 0
rata23 = 0
for x in range(6000,9000):
    if(filter_A2[x]>=0):
        sum23 = sum23 + filter_A2[x]

rata23 = sum23/3000
print ("Rata-Rata Periode Ke 3 Yaitu Detik 60-90 : ",rata23)

sum24 = 0
rata24 = 0
for x in range(9000,12000):
    if(filter_A2[x]>=0):
        sum24 = sum24 + filter_A2[x]

rata24 = sum24/3000
print ("Rata-Rata Periode Ke 4 Yaitu Detik 90-120 : ",rata24)

sum25 = 0
rata25 = 0
for x in range(12000,15000):
    if(filter_A2[x]>=0):
        sum25 = sum25 + filter_A2[x]

rata25 = sum25/3000
print ("Rata-Rata Periode Ke 5 Yaitu Detik 120-150 : ",rata25)

sum26 = 0
rata26 = 0
for x in range(15000,17756):
    if(filter_A2[x]>=0):
        sum26 = sum26 + filter_A2[x]

rata26 = sum26/(17756-15000)
print ("Rata-Rata Periode Ke 6 Yaitu Detik 150-180 : ",rata26)

data_Rata_Rata_A2 = [rata21,rata22,rata23,rata24,rata25,rata26]
ranking_A2 = sorted(data_Rata_Rata_A2)
data_klasifikasi = ['Sepeda Motor 1 Orang','Sepeda Motor 2 Orang','Mobil 1 Orang','Mobil 4-6 Orang','Truck Kosong','Truck Full K

for x in range(0,len(ranking_A2)):
    if(ranking_A2[x]==rata21):
        print("Periode Ke - 1 = 0-30 Detik")
        print("Rata-Rata = ",ranking_A2[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_A2)):
    if(ranking_A2[x]==rata22):
        print("Periode Ke - 2 = 30-60 Detik")
        print("Rata-Rata = ",ranking_A2[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_A2)):
    if(ranking_A2[x]==rata23):
        print("Periode Ke - 3 = 60-90 Detik")
        print("Rata-Rata = ",ranking_A2[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_A2)):
    if(ranking_A2[x]==rata24):
        print("Periode Ke - 4 = 90-120 Detik")
        print("Rata-Rata = ",ranking_A2[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

print("")

for x in range(0,len(ranking_A2)):
    if(ranking_A2[x]==rata25):
        print("Periode Ke - 5 = 120-150 Detik")
        print("Rata-Rata = ",ranking_A2[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

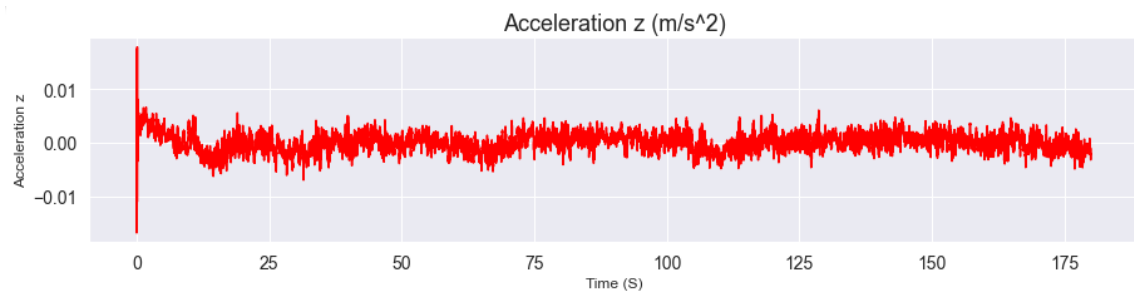
print("")

for x in range(0,len(ranking_A2)):
    if(ranking_A2[x]==rata26):
        print("Periode Ke - 6 = 150-180 Detik")
        print("Rata-Rata = ",ranking_A2[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

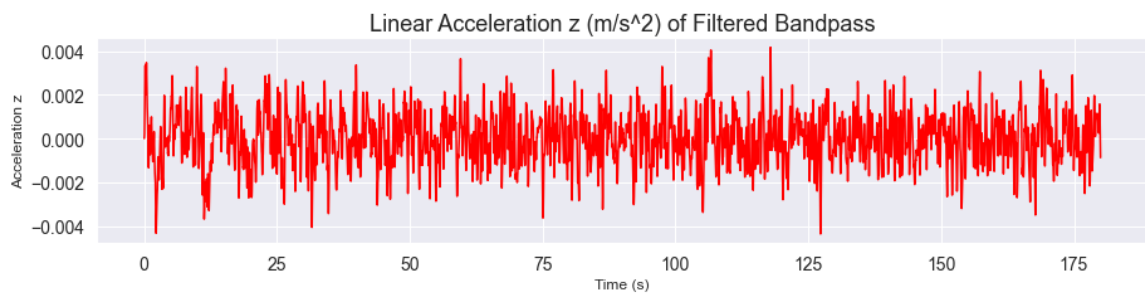
```

Berikut adalah Hasil eksekusi program Linear Acceleration z :

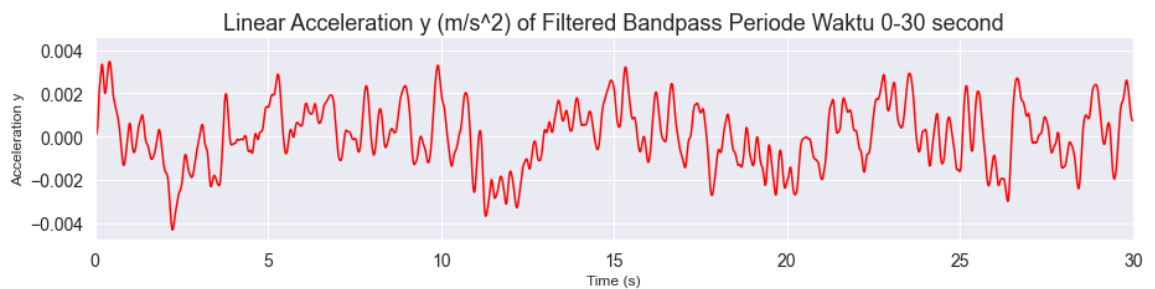
1. Adalah Data Awal dari Linear Acceleration Sumbu Z



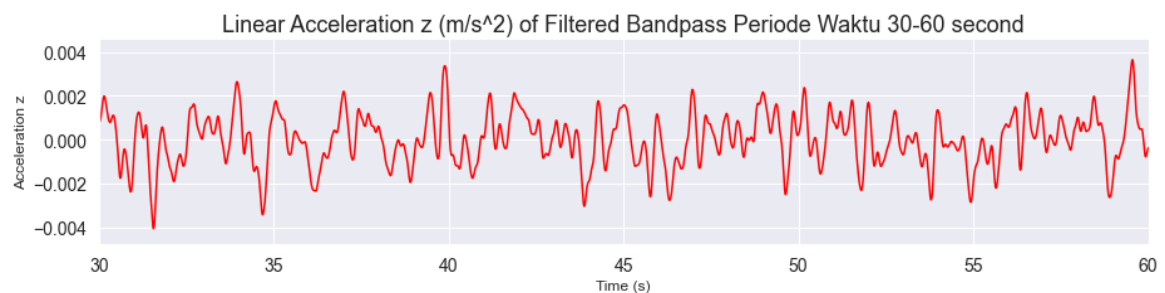
2. Adalah Data Yg Telah Di Filter Menggunakan Bandpass



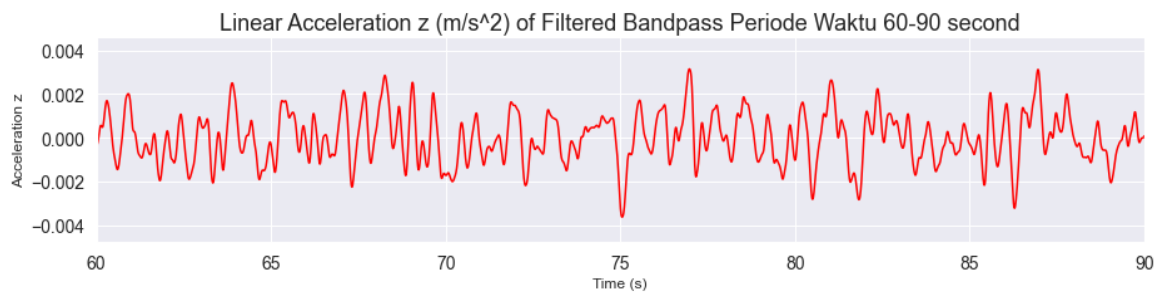
3. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 0-30 & Ini adalah data periode 1



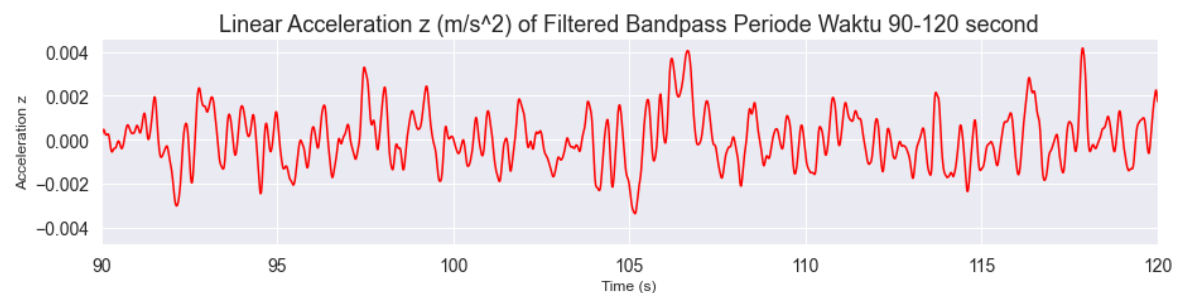
4. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 30-60 & Ini adalah data periode 2



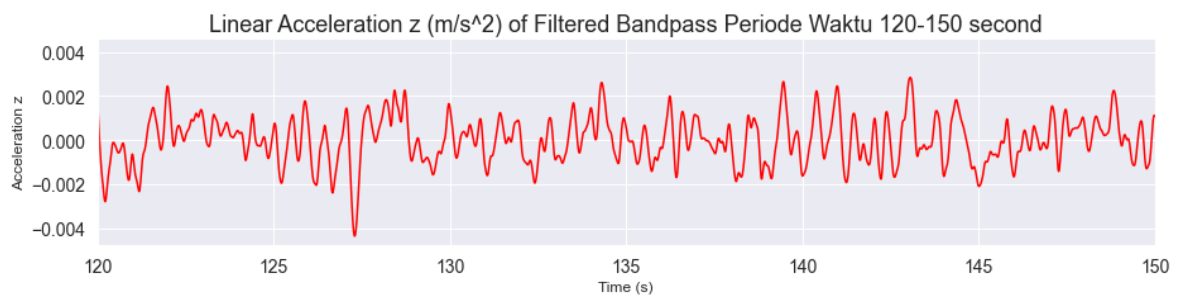
5. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 60-90 & Ini adalah data periode 3



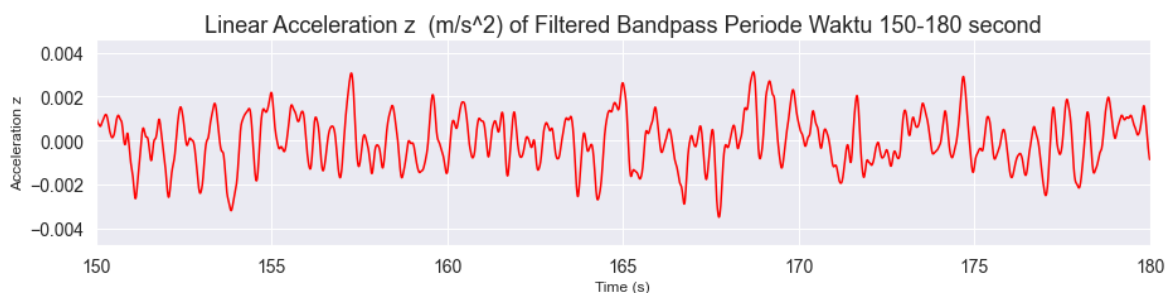
6. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 90-120 & Ini adalah data periode 4



7. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 120-150 & Ini adalah data periode 5



8. Adalah Data Yang Telah Dipotong Sesuai Periode dari Detik 150-180 & Ini adalah data periode 6



9. Dan Ini Adalah Hasil Rata Rata Dari Periode ke 1 hingga period ke 6

```
===== Linear Acceleration z Clasification =====
Rata-Rata Periode Ke 1 Yaitu Detik 0-30      : 0.0005987225828874627
Rata-Rata Periode Ke 2 Yaitu Detik 30-60     : 0.0004513411494919367
Rata-Rata Periode Ke 3 Yaitu Detik 60-90     : 0.0004424020849218419
Rata-Rata Periode Ke 4 Yaitu Detik 90-120    : 0.0004755507285605102
Rata-Rata Periode Ke 5 Yaitu Detik 120-150   : 0.00042573827997504596
Rata-Rata Periode Ke 6 Yaitu Detik 150-180   : 0.0004743693941934892
```

10. Lalu Data Tsb Akan di ranking dengan menggunakan sorting secara ascending lalu dari data ascending tsb dicocokkan dengan data klasifikasi kendaraan dan menghasilkan berupa urutan data periode .

```

Periode Ke - 1      = 0-30 Detik
Rata-Rata          = 0.0005987225828874627
Prediksi Kendaraan = Truck Full Kapasitas

Periode Ke - 2      = 30-60 Detik
Rata-Rata          = 0.0004513411494919367
Prediksi Kendaraan = Mobil 1 Orang

Periode Ke - 3      = 60-90 Detik
Rata-Rata          = 0.0004424020849218419
Prediksi Kendaraan = Sepeda Motor 2 Orang

Periode Ke - 4      = 90-120 Detik
Rata-Rata          = 0.0004755507285605102
Prediksi Kendaraan = Truck Kosong

Periode Ke - 5      = 120-150 Detik
Rata-Rata          = 0.00042573827997504596
Prediksi Kendaraan = Sepeda Motor 1 Orang

Periode Ke - 6      = 150-180 Detik
Rata-Rata          = 0.0004743693941934892
Prediksi Kendaraan = Mobil 4-6 Orang

```

Gambar Diatas Adalah Menunjukkan hasil klasifikasi dengan ide yang saya jelaskan diatas yang diambil dari data Linear Acceleration Sumbu Z

Terdapat Hasil Yang akan Dirata ratakan setiap linear acceleration X,Y,Z pada periode waktu ke 1 hingga ke 6 dan di klasifikasi dengan data klasifikasi kendaraan dan ini adalah code program dan hasil eksekusinya

Code Program Rata-Rata Sumbu X,Y,Z pada periode 1-6 dan mengklasifikasi data kendaraan

```

data_Rata_Rata_Axyz = [(sum([ratax1,ratay1,rataz1])/3),(sum([ratax2,ratay2,rataz2])/3),(sum([ratax3,ratay3,rataz3])/3),
                       (sum([ratax4,ratay4,rataz4])/3),(sum([ratax5,ratay5,rataz5])/3),(sum([ratax6,ratay6,rataz6])/3)]
ranking_Axyz = sorted(data_Rata_Rata_Axyz)
data_klasifikasi = ['Sepeda Motor 1 Orang', 'Sepeda Motor 2 Orang', 'Mobil 1 Orang', 'Mobil 4-6 Orang', 'Truck Kosong', 'Truck Full K

for x in range(0,len(ranking_Axyz)):
    if(ranking_Axyz[x]==(sum([ratax1,ratay1,rataz1])/3)):
        print("Periode Ke - 1      = 0-30 Detik")
        print("Rata-Rata          = ",ranking_Axyz[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

    print("")

for x in range(0,len(ranking_Axyz)):
    if(ranking_Axyz[x]==(sum([ratax2,ratay2,rataz2])/3)):
        print("Periode Ke - 2      = 30-60 Detik")
        print("Rata-Rata          = ",ranking_Axyz[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

    print("")

for x in range(0,len(ranking_Axyz)):
    if(ranking_Axyz[x]==(sum([ratax3,ratay3,rataz3])/3)):
        print("Periode Ke - 3      = 60-90 Detik")
        print("Rata-Rata          = ",ranking_Axyz[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

    print("")

for x in range(0,len(ranking_Axyz)):
    if(ranking_Axyz[x]==(sum([ratax4,ratay4,rataz4])/3)):
        print("Periode Ke - 4      = 90-120 Detik")
        print("Rata-Rata          = ",ranking_Axyz[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

    print("")

for x in range(0,len(ranking_Axyz)):
    if(ranking_Axyz[x]==(sum([ratax5,ratay5,rataz5])/3)):
        print("Periode Ke - 5      = 120-150 Detik")
        print("Rata-Rata          = ",ranking_Axyz[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

    print("")

for x in range(0,len(ranking_Axyz)):
    if(ranking_Axyz[x]==(sum([ratax6,ratay6,rataz6])/3)):
        print("Periode Ke - 6      = 150-180 Detik")
        print("Rata-Rata          = ",ranking_Axyz[x])
        print("Prediksi Kendaraan = ",data_klasifikasi[x])

```

Dan Berikut Hasil Eksekusinya terhadap rata-rata dan periode 1-6 pada klasifikasi kendaraan data Linear Acceleration X,Y,Z

```
Periode Ke - 1      = 0-30 Detik
Rata-Rata           = 0.0002676207575683902
Prediksi Kendaraan = Truck Full Kapasitas

Periode Ke - 2      = 30-60 Detik
Rata-Rata           = 0.00021941137278585037
Prediksi Kendaraan = Mobil 1 Orang

Periode Ke - 3      = 60-90 Detik
Rata-Rata           = 0.00021078257834514875
Prediksi Kendaraan = Sepeda Motor 2 Orang

Periode Ke - 4      = 90-120 Detik
Rata-Rata           = 0.00022339781462456837
Prediksi Kendaraan = Mobil 4-6 Orang

Periode Ke - 5      = 120-150 Detik
Rata-Rata           = 0.00020865769937954888
Prediksi Kendaraan = Sepeda Motor 1 Orang

Periode Ke - 6      = 150-180 Detik
Rata-Rata           = 0.00023499592120700507
Prediksi Kendaraan = Truck Kosong
```

Kesimpulan : Kesimpulan Pada code program keseluruhan adalah aplikasi yang dibuat akan menghasilkan output berupa klasifikasi kendaraan yang melewati yaitu dengan beberapa periode per detik jika di periode ke 1 maka ada banyak kendaraan yang lewat yaitu berjenis "Truck Full Kapasitas" dengan penggabungan data rata-rata dari linear acceleration X,Y,Z Yang telah di filter menggunakan bandpass filter dan hasil yang ditunjukkan pada FFT adalah nilai frekuensi natural yang rendah yaitu range 1-10 Hz dapat di prediksi bahwa nilai data Periode Ke 1 adalah Waktu dimana Nilai mendapatkan frekuensi yang tinggi maka bisa disebutkan bahwa Periode 1 adalah puncak dari getaran yang bisa dianggap dengan klasifikasi kendaraan yaitu berjenis "Truck Full Kapasitas" dan penjelasan periode ke 2 hingga ke 6 adalah berupa penjelasan sama seperti penjelasan periode 1 namun nilai periode ke 2 hingga ke 6 adalah lebih rendah dibanding nilai periode ke 1.

Dan Ini Adalah Source Code Program Jika Ingin Dijalankan Menggunakan Jupyter Notebook atau menggunakan Python 3

<https://github.com/stanaedro/UTS-IOT-StanaEdroSwargara>