PERTEMUAN 9

Density Estimation Menggunakan Python

Nama : Rangga Pebrianto
NIM : G6601231006

Kepekatan Peluang

Suatu peubah acak x memiliki peluang distribusi p(x). Apabila peubah acak tersebut kontinyu, maka nilai peluang dapat dihitung dengan menggunakan fungsi kepakatan peluang (probabilty density function atau yang sering dikenal dengan istilah PDF. Bentuk fungsi kepekatan peluang pada semua domain peubah acak tersebut mengacu pada dirtibusi peluang yang beragam apakah itu normal, eksponensial dan lainnya.

Mengetahui distribusi probabilitas untuk variabel acak dapat membantu menghitung distribusi, seperti mean dan varians, juga dapat berguna mempertimbangkan kejadian tertentu, seperti menentukan apakah pengamatan tidak mungkin atau tidak dengan adanaya suatu pencilan pada data yang kita miliki.

TUJUAN PRAKTIKUM

Pada praktikum Pertemuan 10 akan menerapkan beberapa konsep antara lain :

1. Kepekatan peluang
2. Kepekatan dengan menggunakan histogram
3. Estimasi kepekatan parametrik
4. Estimasi kepekatan nonparametrik

Keempat konsep tersebut diterapkan pada sebuah permasalahan dengan menggunakan tools / bahasa pemrograman Python.

1. Kepekatan Peluang

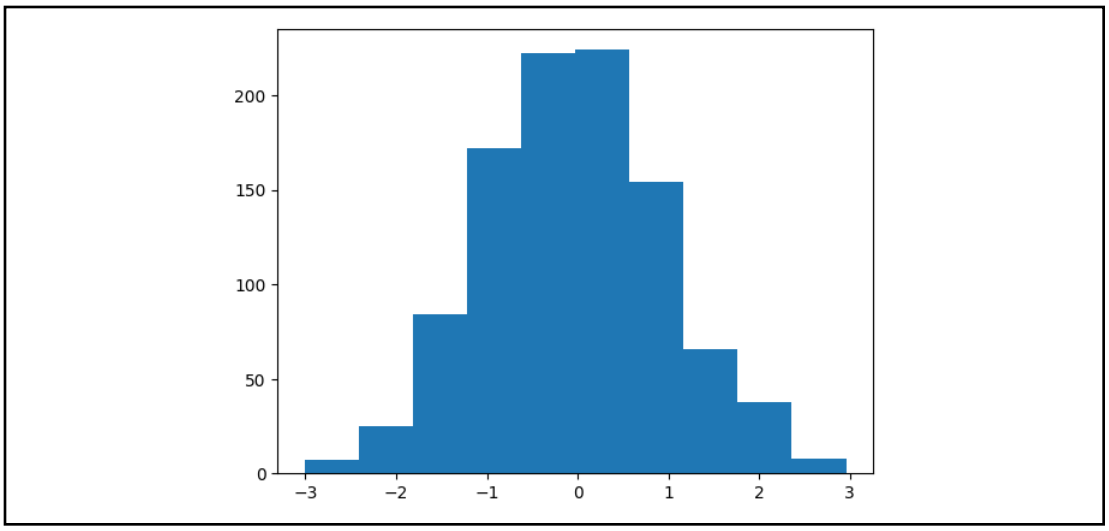
Dikarenakan kita tidak mengatahui peluang distriusi semua variabel acak, sehingga yang bisa kita lakukan adalah menentukan peluang distribusi tersebut melalui observasi atau yang lebih dikenal dengan melakukan estimasi peluang kepekatan. Hal ini lebih dikenal dengan *"density estimation".*

Langkah pertama yang dilakukan untuk melihat kepekatan suatu observasi pada sampel acak ialah dengan histogram sederhana. Dari histogram, kita dapat mengidentifikasi peluang distribusi probabilitas yang secara umum dapat dipahami seperti distribusi normal. Jika tidak, kita harus menyesuaikan model untuk melakukan estimasi distribusi.

2. Kepekatan Peluang dengan Histogram

```
from matplotlib import pyplot
from numpy.random import normal
# generate a sample
```

```
sample = normal(size=1000)
# plot a histogram of the sample
pyplot.hist(sample, bins=10)
pyplot.show()
```

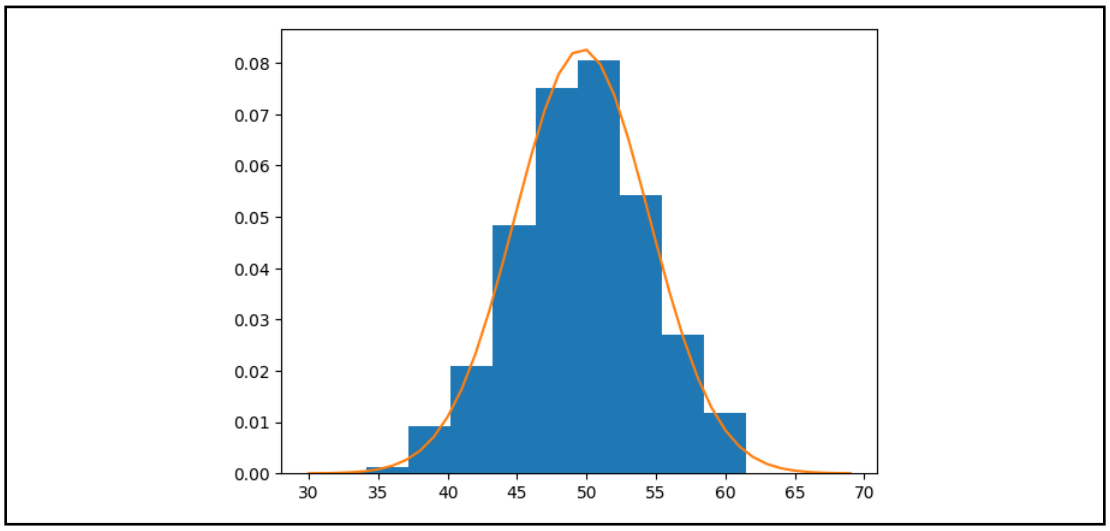


a. Estimasi Kepekatan Parameterik

Bentuk histogram dari suatu sampel acak akan cocok dengan distribusi probabilitas yang kita ketahui. Mengetahui peluang distribusi ini akan membantu mengenali/identifikasi distribusi jenis apa dari data yang kita miliki. Setelah mampu melakukan identifkasi, kita dapat melakukan estimasi kepadatan dengan memperkirakan parameter distribusi dari data acak yang digunakan. Sebagai contoh, distribusi normal memiliki dua parameter, yaitu mean dan standar deviasi. Distribusi yang berbeda, akan memiliki parameter yang berbeda juga.

```
# contoh parametric probability density estimation
# library yang digunakan

from matplotlib import pyplot
from numpy.random import normal
from numpy import mean
from numpy import std
from scipy.stats import norm

# membangkitkan sample
sample = normal(loc=50, scale=5, size=1000)

# menghitung parameters
sample_mean = mean(sample)
sample_std = std(sample)
print('Mean=%.3f, Standard Deviation=%.3f' % (sample_mean,
sample_std))
```

```
# mendefinisikan distribusi
dist = norm(sample_mean, sample_std)
values = [value for value in range(30, 70)]
probabilities = [dist.pdf(value) for value in values]

# plotting histogram and pdf
pyplot.hist(sample, bins=10, density=True)
pyplot.plot(values, probabilities)
pyplot.show()
```



3. Estimasi Kepekatan Nonparameterik

Pada beberapa kasus, sampel data tidak menyerupai distribusi probabilitas yang umum digunakan atau tidak mudah dibuat sesuai dengan distribusi tersebut. Hal ini sering terjadi ketika data memiliki dua puncak (distribusi bimodal) atau banyak puncak (distribusi multimodal). Oleh karena itu, tidak bisa menggunakan pendekatan estimasi parametrik sehingga digunakan pendekatan lain yaitu kernel density estimation atau yang dikenal dengan istilah Parzen Window.

Contoh dari kondisi ini disajikan sebagai berikut:

Library yang diguanakan

```
from matplotlib import pyplot
from numpy.random import normal
from numpy import hstack
from numpy import asarray
from numpy import exp
from sklearn.neighbors import KernelDensity
```

Ilustrasi kondisi data bimodal dengan dua ditribusi yang berbeda

```
# contoh kondisi bimodal data
from matplotlib import pyplot
from numpy.random import normal
from numpy import hstack
# membangkitkan sample
sample1 = normal(loc=20, scale=5, size=300)
sample2 = normal(loc=40, scale=5, size=700)
sample = hstack((sample1, sample2))
# plot histogram
pyplot.hist(sample, bins=50)
pyplot.show()
```

Untuk melakukan *fitting* terhadap sample tersebut, kita gunakan fungsi kernal density estimation

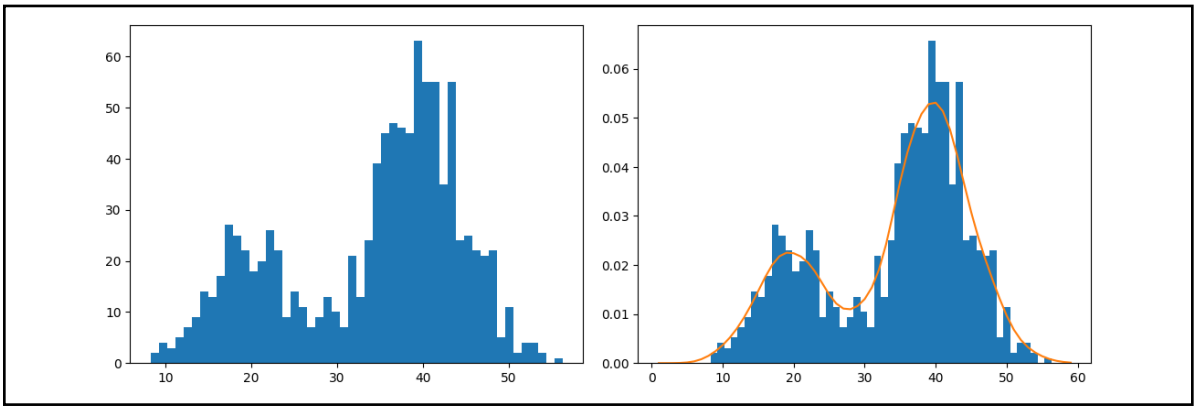
```
model = KernelDensity(bandwidth=2, kernel='gaussian')
sample = sample.reshape((len(sample), 1))
model.fit(sample)
```

Sebagai contoh yang digunakan adalah fungsi kepekatan kernal yang terdapa di dalam library scikit. Langkah berikutnya adalah menghitung peluang sample pada range tertentu

```
# sample probabilities for a range of outcomes
values = asarray([value for value in range(1, 60)])
values = values.reshape((len(values), 1))
probabilities = model.score_samples(values)
probabilities = exp(probabilities)
```

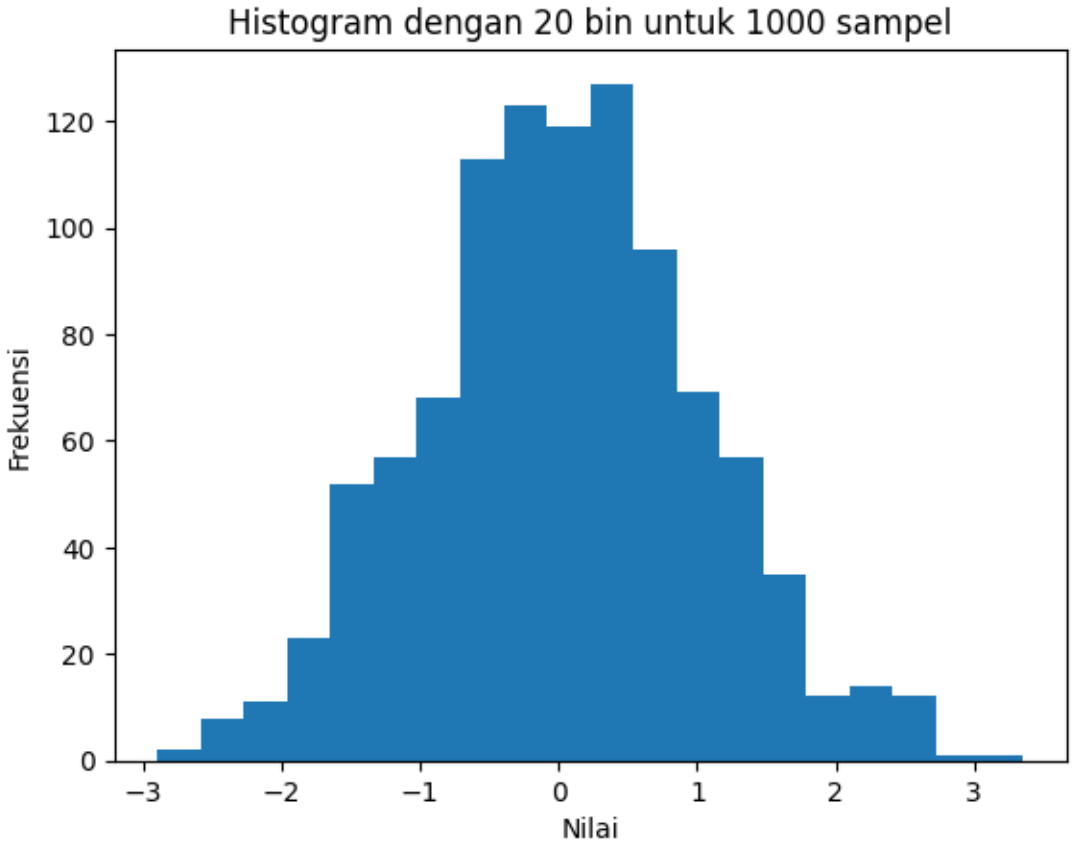
Setelah menentukan peluang sample tahap berikutya adalah melakukan plotting ke dalam kurva.

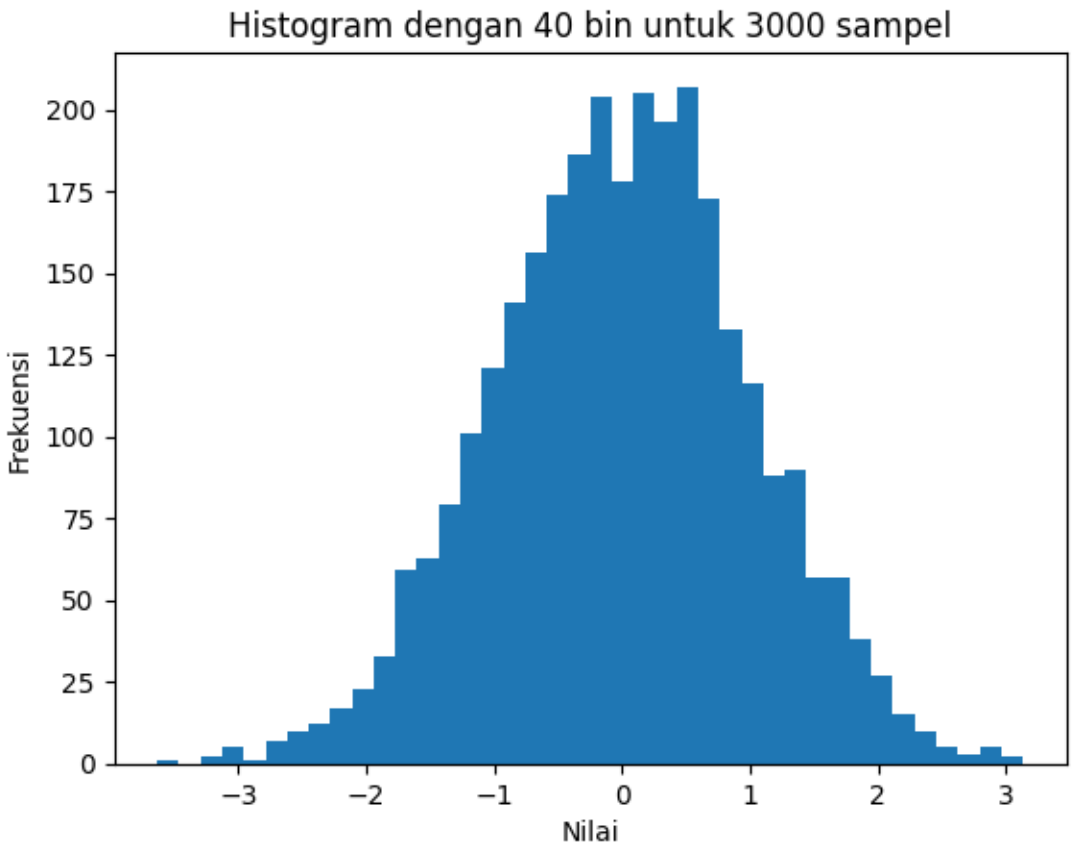
```
# plot the histogram and pdf
pyplot.hist(sample, bins=50, density=True)
pyplot.plot(values[:], probabilities)
pyplot.show()
```

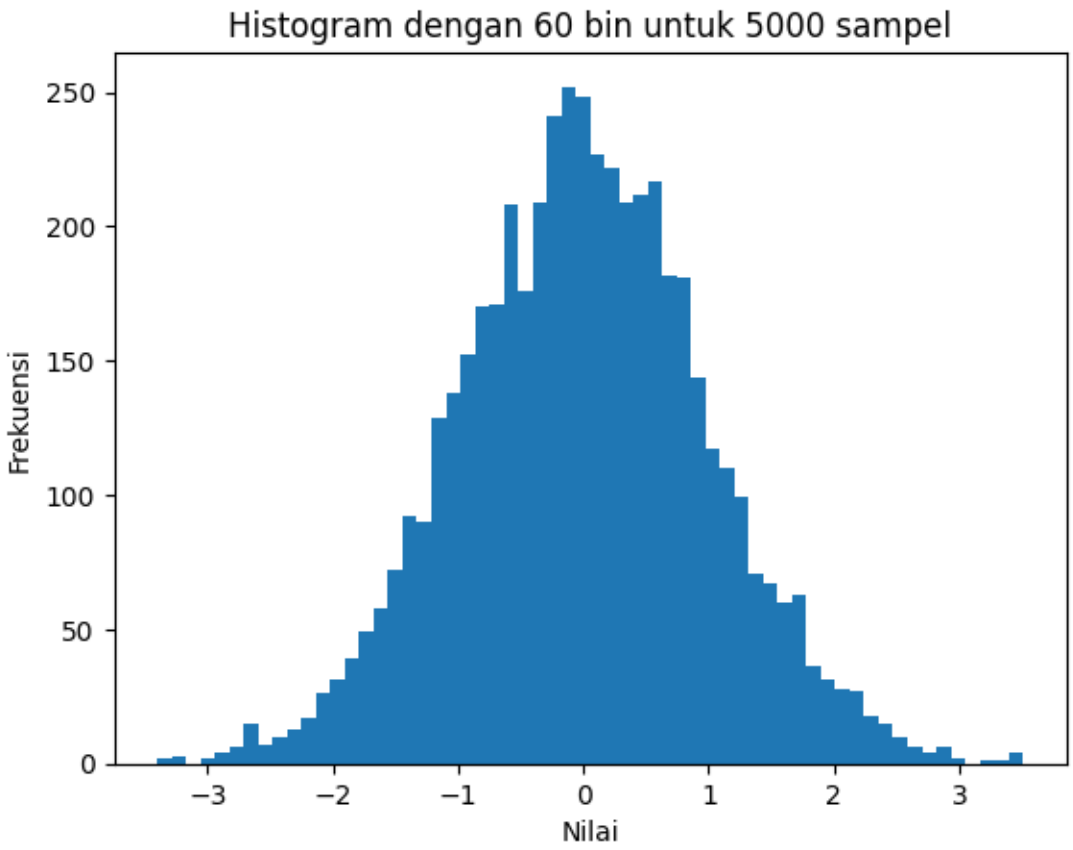


TUGAS PRAKTIKUM

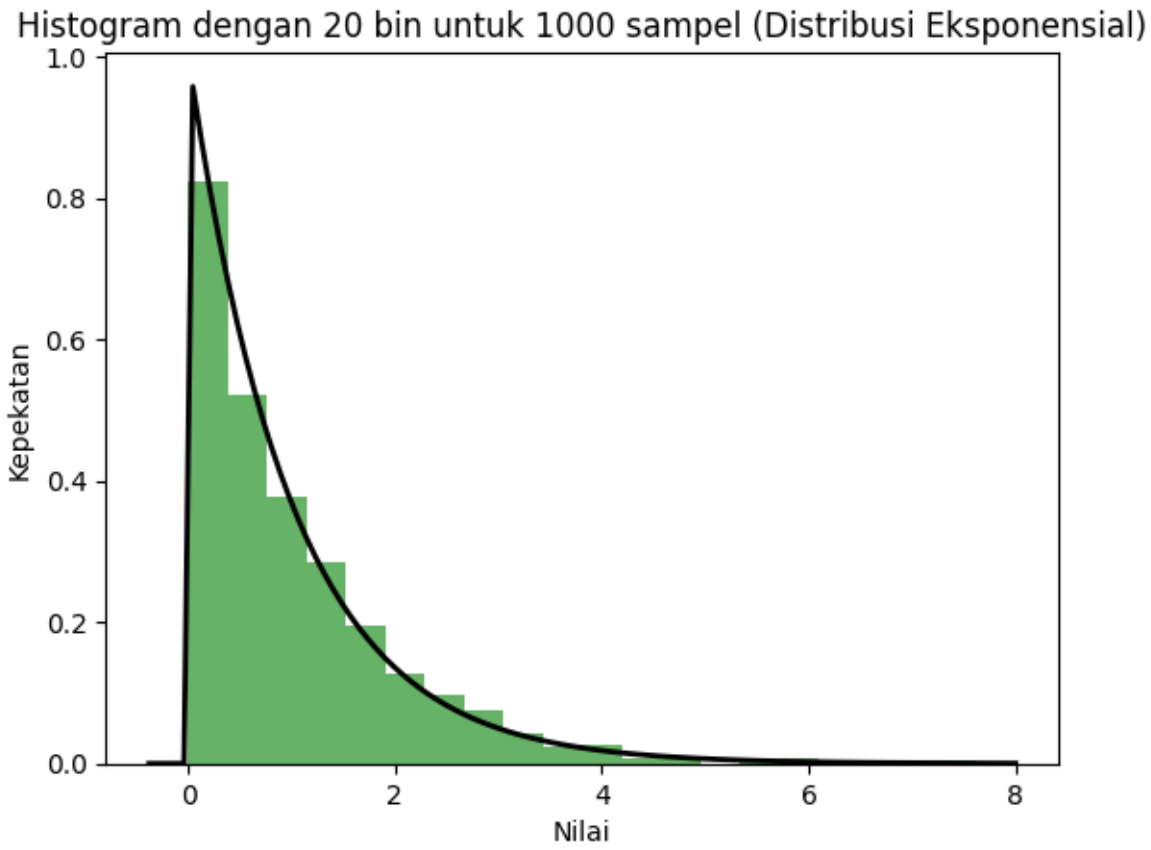
1. Lakukan beberapa modifikasi pada nilai bin dan sampel serta visualisasikan histogram tersebut dan sajikan pada box di bawah ini.
   Sample : 1000, 3000, 5000
   bin : 20, 40, 60

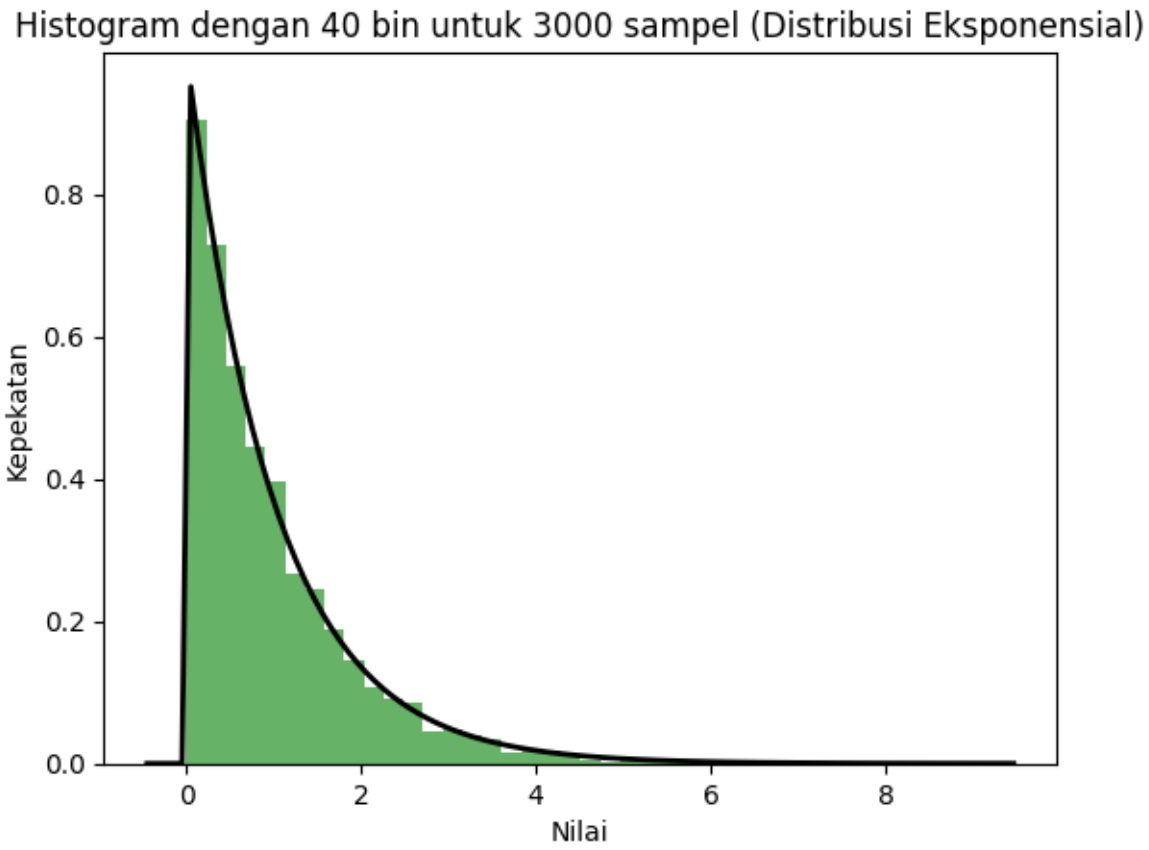


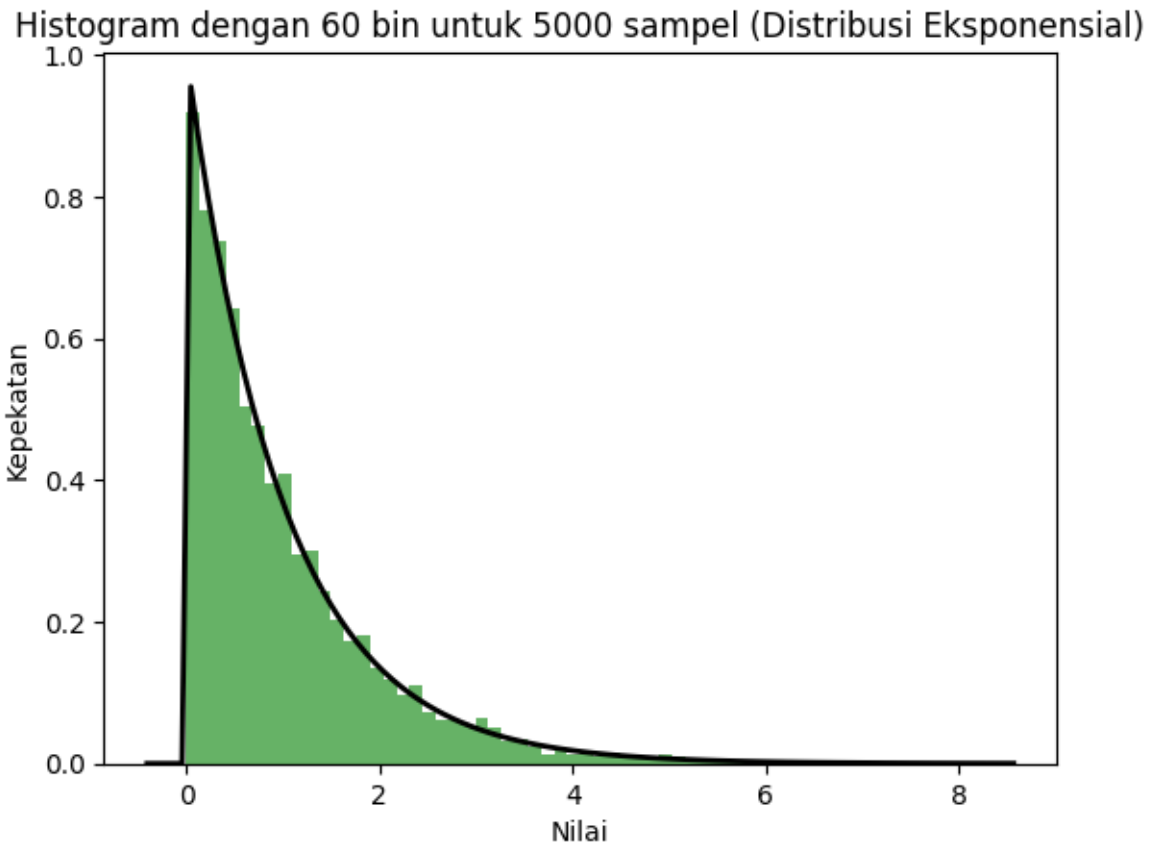

Histogram dengan 40 bin untuk 3000 sampel
Frekuensi
Nilai
0
25
50
75
100
125
150
175
200
−3
−2
−1
0
1
2
3



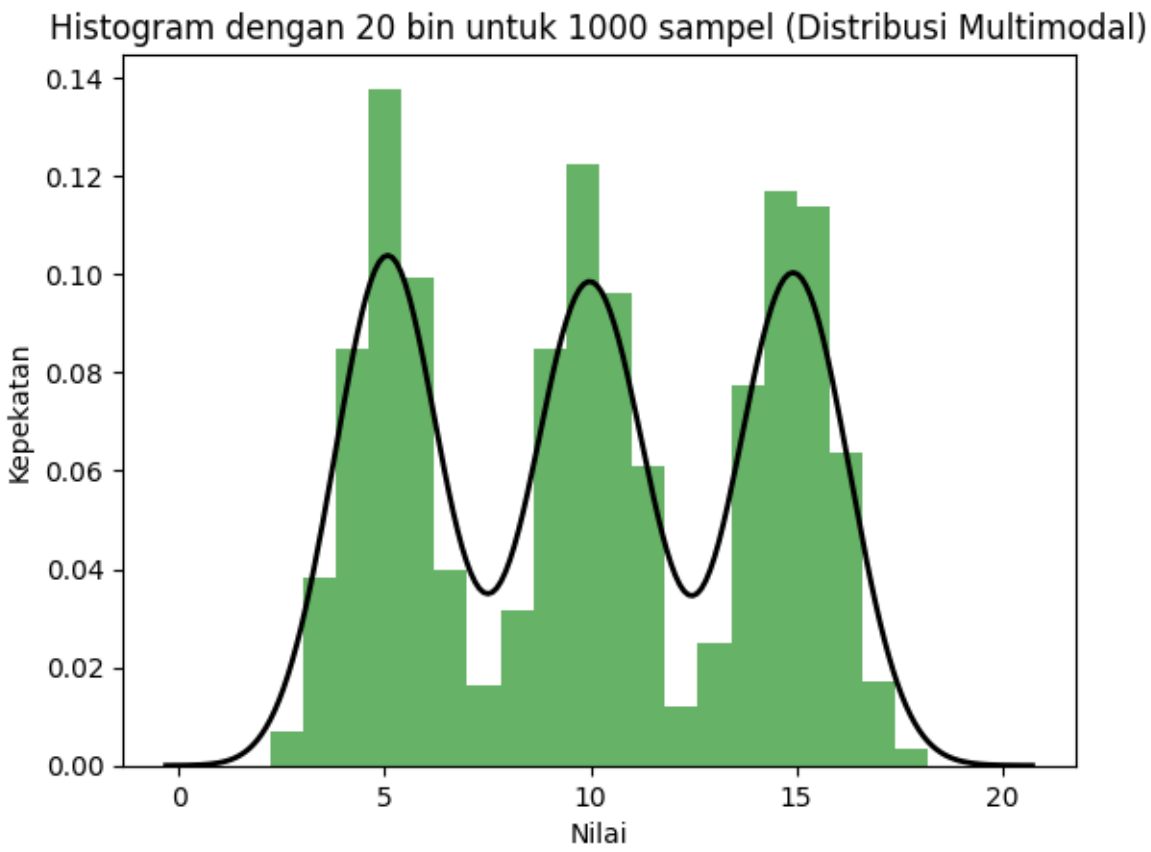
2. Pada estimasi kepekatan parametrik, cari distribusi lain selain distribusi normal dan lakukan visualiasi

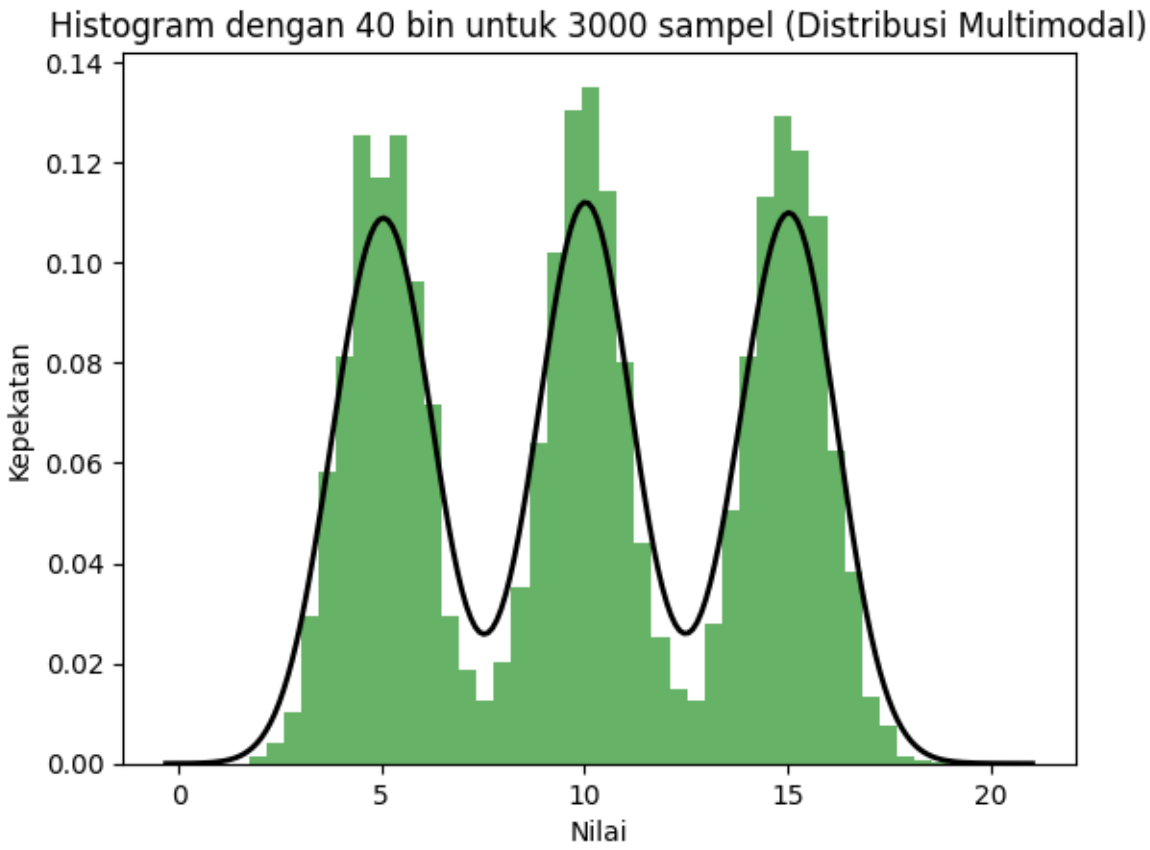

Histogram dengan 20 bin untuk 1000 sampel (Distribusi Eksponensial)
Kepekatan
Nilai
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8

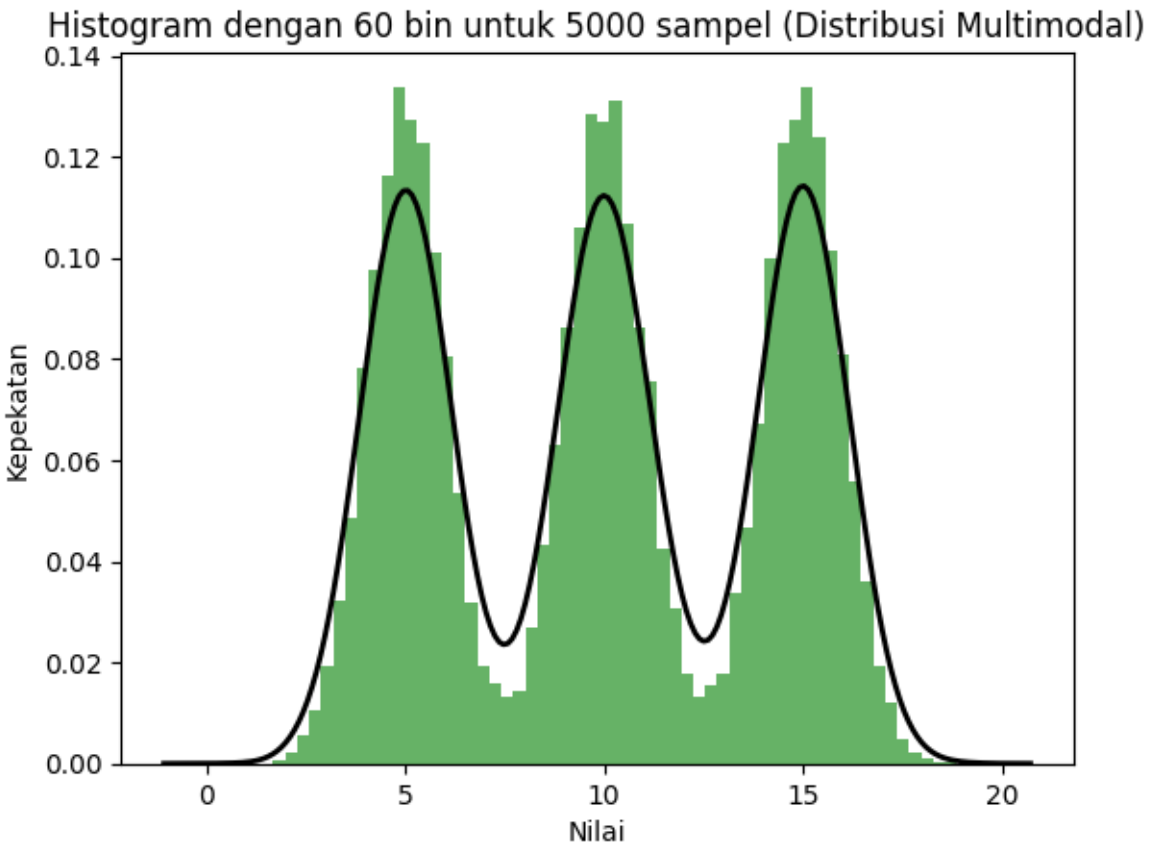

Histogram dengan 40 bin untuk 3000 sampel (Distribusi Eksponensial)
Kepekatan
0.0
0.2
0.4
0.6
0.8
0
2
4
6
8
Nilai



3. Buat histogram untuk contoh pembangkitan data dengan jumlah peak = 3 (multimodal) kemudian gunakan fungsi kernel density estimation dan lakukan visualisasi


Histogram dengan 20 bin untuk 1000 sampel (Distribusi Multimodal)
Kepekatan
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
Nilai


Histogram dengan 40 bin untuk 3000 sampel (Distribusi Multimodal)
Kepekatan
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
Nilai

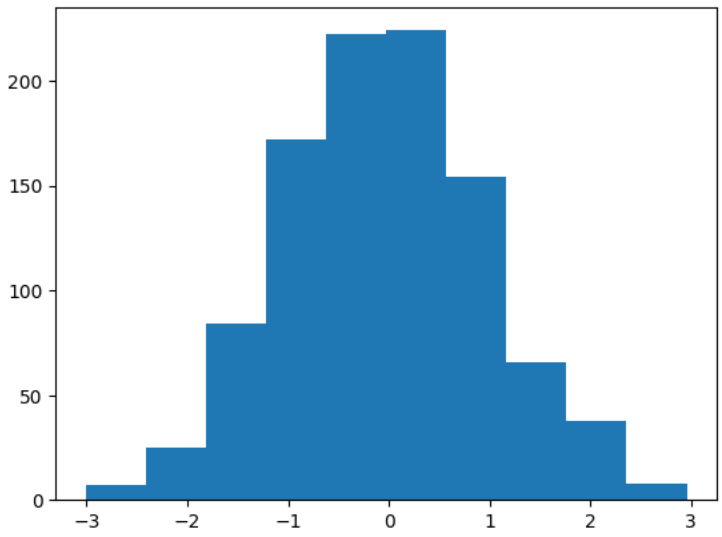


DAFTAR PUSTAKA

1. Richert W & Coelho LP. *Builidng Machine Learning System with Python*. 2013. Packt Publising. Birmingham, UK.
2. https://machinelearningmastery.com/probability-density-estimation/

Kepekatan Peluang dengan Histogram

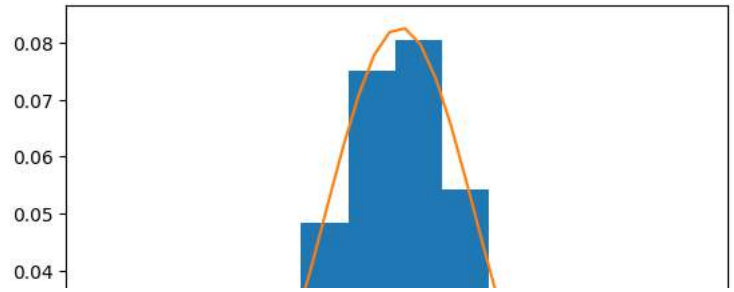
```
from matplotlib import pyplot
from numpy.random import normal
# generate a sample
sample = normal(size=1000)
# plot a histogram of the sample
pyplot.hist(sample, bins=10)
pyplot.show()
```



Estimasi Kepekatan Parameterik

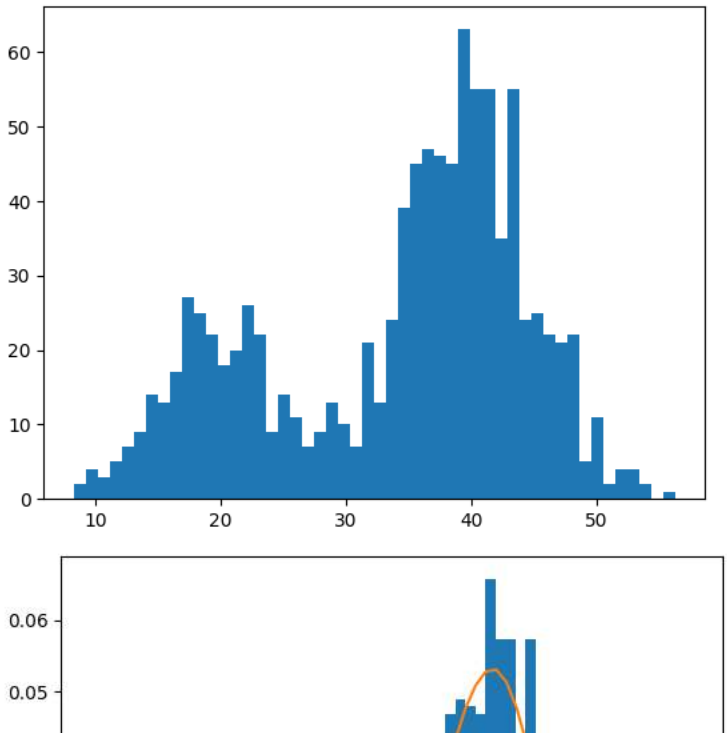
```
# contoh parametric probability density estimation
# library yang digunakan

from matplotlib import pyplot
from numpy.random import normal
from numpy import mean
from numpy import std
from scipy.stats import norm

# membangkitkan sample
sample = normal(loc=50, scale=5, size=1000)

# menghitung parameters
sample_mean = mean(sample)
sample_std = std(sample)
print('Mean=%.3f, Standard Deviation=%.3f' % (sample_mean, sample_std))

# mendefinisikan distribusi
dist = norm(sample_mean, sample_std)
values = [value for value in range(30, 70)]
probabilities = [dist.pdf(value) for value in values]

# plotting histogram and pdf
pyplot.hist(sample, bins=10, density=True)
pyplot.plot(values, probabilities)
pyplot.show()
```

```
Mean=49.682, Standard Deviation=4.823
```



Estimasi Kepekatan Nonparameterik

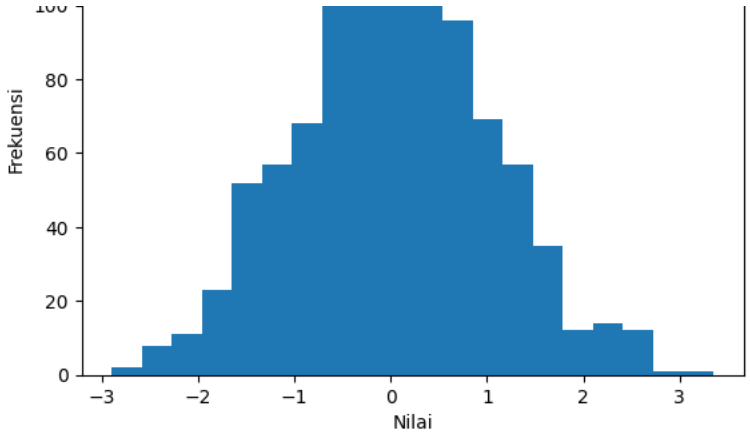
```
# Library yang diguanakan
from matplotlib import pyplot
from numpy.random import normal
from numpy import hstack
from numpy import asarray
from numpy import exp
from sklearn.neighbors import KernelDensity


#Ilustrasi kondisi data bimodal dengan dua ditribusi yang berbeda


# contoh kondisi bimodal data
from matplotlib import pyplot
from numpy.random import normal
from numpy import hstack
# membangkitkan sample
sample1 = normal(loc=20, scale=5, size=300)
sample2 = normal(loc=40, scale=5, size=700)
sample = hstack((sample1, sample2))
# plot histogram
pyplot.hist(sample, bins=50)
pyplot.show()

#Untuk melakukan fitting terhadap sample tersebut, kita gunakan fungsi kernal density estimation

model = KernelDensity(bandwidth=2, kernel='gaussian')
sample = sample.reshape((len(sample), 1))
model.fit(sample)

#Sebagai contoh yang digunakan adalah fungsi kepekatan kernal yang terdapa di dalam library scikit. Langkah berikutnya adalah menghitung pelua

# sample probabilities for a range of outcomes
values = asarray([value for value in range(1, 60)])
values = values.reshape((len(values), 1))
probabilities = model.score_samples(values)
probabilities = exp(probabilities)

 #Setelah menentukan peluang sample tahap berikutya adalah melakukan plotting ke dalam kurva.
# plot the histogram and pdf
pyplot.hist(sample, bins=50, density=True)
pyplot.plot(values[:], probabilities)
pyplot.show()
```



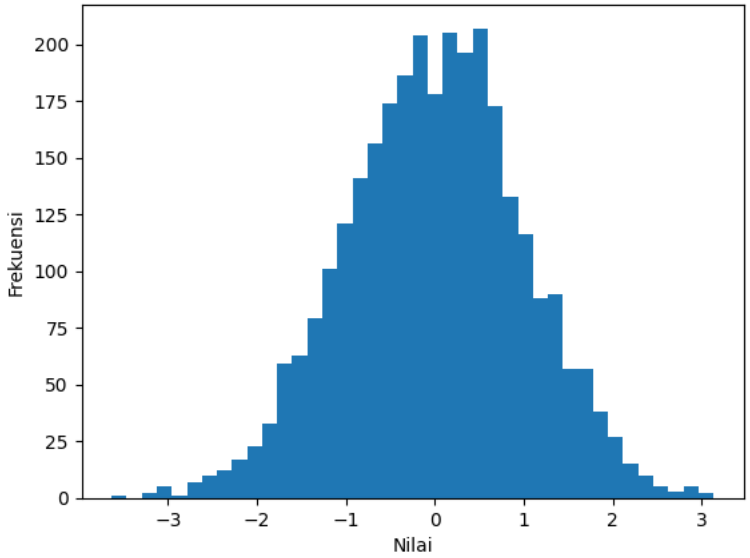
Lakukan beberapa modifikasi pada nilai bin dan sampel serta visualisasikan histogram tersebut dan sajikan pada box di bawah ini.

Sample : 1000, 3000, 5000 bin : 20, 40, 60


```
from matplotlib import pyplot
from numpy.random import normal

# Mengubah nilai sampel dan jumlah bin
samples = [1000, 3000, 5000]
bins = [20, 40, 60]

# Loop melalui nilai sampel dan jumlah bin
for i in range(len(samples)):
    # Generate sample data
    sample_data = normal(size=samples[i])

    # Plot histogram
    pyplot.hist(sample_data, bins=bins[i])
    pyplot.title(f'Histogram dengan {bins[i]} bin untuk {samples[i]} sampel')
    pyplot.xlabel('Nilai')
    pyplot.ylabel('Frekuensi')
    pyplot.show()
```



Histogram dengan 40 bin untuk 3000 sampel

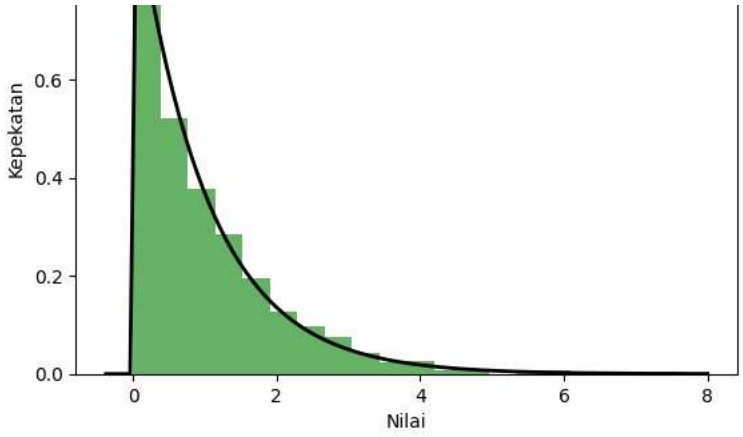


Histogram dengan 60 bin untuk 5000 sampel

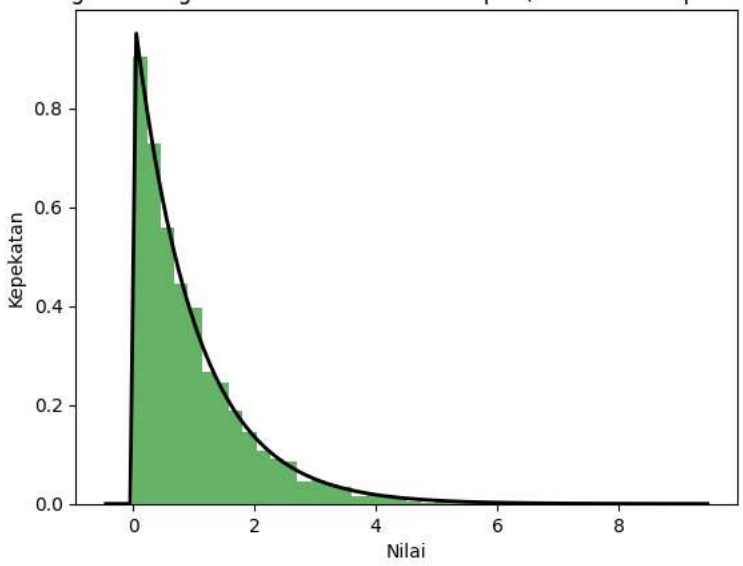
Pada estimasi kepekatan parametrik, cari distribusi lain selain distribusi normal dan lakukan visualiasi

```
from matplotlib import pyplot
from numpy.random import exponential
import numpy as np
from scipy.stats import expon  # Import distribusi eksponensial dari SciPy

# Mengubah nilai sampel dan jumlah bin
samples = [1000, 3000, 5000]
bins = [20, 40, 60]

# Loop melalui nilai sampel dan jumlah bin
for i in range(len(samples)):
    # Generate sample data dengan distribusi eksponensial
    sample_data = exponential(scale=1, size=samples[i])  # Parameter scale bisa diubah sesuai kebutuhan

    # Plot histogram
    pyplot.hist(sample_data, bins=bins[i], density=True, alpha=0.6, color='g')  # density=True agar area di bawah histogram = 1
    pyplot.title(f'Histogram dengan {bins[i]} bin untuk {samples[i]} sampel (Distribusi Eksponensial)')
    pyplot.xlabel('Nilai')
    pyplot.ylabel('Kepekatan')

    # Menambahkan kurva distribusi eksponensial pada histogram
    xmin, xmax = pyplot.xlim()
    x = np.linspace(xmin, xmax, 100)
    p = expon.pdf(x, scale=1)  # Distribusi eksponensial dengan parameter scale=1
```

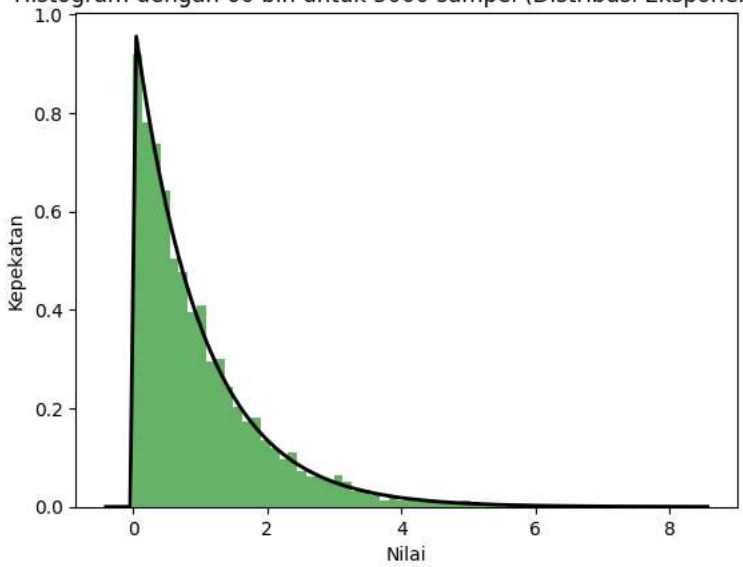
```
pyplot.plot(x, p, 'k', linewidth=2)

pyplot.show()
```



Histogram dengan 40 bin untuk 3000 sampel (Distribusi Eksponensial)



Histogram dengan 60 bin untuk 5000 sampel (Distribusi Eksponensial)



Buat histogram untuk contoh pembangkitan data dengan jumlah peak = 3 (multimodal) kemudian gunakan fungsi kernel density estimation dan lakukan visualisasi

```
from matplotlib import pyplot
from numpy.random import normal
import seaborn as sns
import numpy as np

# Mengubah nilai sampel dan jumlah bin
samples = [1000, 3000, 5000]
bins = [20, 40, 60]

# Loop melalui nilai sampel dan jumlah bin
for i in range(len(samples)):
    # Generate sample data dengan distribusi normal multimodal
    data_peak1 = normal(loc=5, scale=1, size=samples[i])
    data_peak2 = normal(loc=10, scale=1, size=samples[i])
    data_peak3 = normal(loc=15, scale=1, size=samples[i])
    sample_data = np.concatenate([data_peak1, data_peak2, data_peak3])

    # Plot histogram
    pyplot.hist(sample_data, bins=bins[i], density=True, alpha=0.6, color='g')
    pyplot.title(f'Histogram dengan {bins[i]} bin untuk {samples[i]} sampel (Distribusi Multimodal)')
    pyplot.xlabel('Nilai')
    pyplot.ylabel('Kepekatan')

    # Menambahkan kurva KDE pada histogram menggunakan Seaborn
    sns.kdeplot(sample_data, color='k', linewidth=2)

    pyplot.show()
```