Nama: Mohammad Afif R. lingkeh **NIM:** **065002400002**		**MODUL 5** **Nama Dosen:** **Drs. Joko Riyono, M.Si.**
Hari/Tanggal: **Jumat, 2 mei 2025**	**Praktikum Probabilitas dan Statistika**	**Nama Asisten Labratorium:** **Tarum Widyasti Pertiwi** **(064002200027)** **Kharisma Maulida Saara** **(064002200024)**

Probabilitas Peubah Acak Normal dan Eksponensial

1. Teori Singkat

Peubah acak (random variable_ adalah variabel yang nilainya didapatkan dari nilai numerik suatu kejadian. Peubah acak juga merupakan fungsi yang memetakan set dari hasil-hasil yang mungkin dari suatu percobaan ke dalam angka. Terdapat dua jenis peubah acak yaitu diskrit dan kontinu. Peubah acak diskrit adalah jenis peubah acak di mana ruang sampelnya terdiri dari seperangkat nilai yang terbatas atau terhitung atau disebut juga dalam ruang bilangan cacah. Dua distribusi peluang peubah acak diskrit yang umum adalah distribusi binomial dan distribusi Poisson. Sedangkan peubah acak kontinu merepresentasikan hasil yang berasal dari suatu rentang nilai bilangan real.

Distribusi Normal

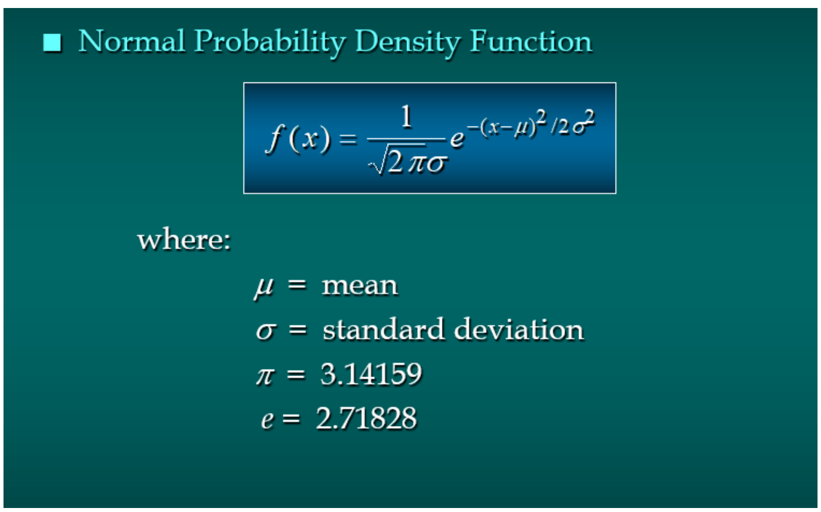
Distribusi normal dikenal sebagai distribusi Gaussian, adalah salah satu distribusi probabilitas kontinu yang paling penting dan sering digunakan dalam statistika. Distribusi ini memiliki bentuk kurva seperti lonceng simetris yang ditentukan oleh dua parameter, yaitu rata-rata (μ) dan standar deviasi (σ).

Karakteristik Distribusi Probabilitas Normal

1. Bentuk kurva normal sering digambarkan sebagai kurva berbentuk lonceng.

2. Dua parameter, m (rata-rata) dan s (standar deviasi), menentukan lokasi dan bentuk distribusi.
3. Titik tertinggi pada kurva normal berada pada rata-rata, yang juga merupakan median dan modus. Rata-rata bisa memiliki nilai numerik apa pun: negatif, nol, atau positif.
4. Kurva normal bersifat simetris.
5. Standar deviasi menentukan lebar kurva: nilai yang lebih besar menghasilkan kurva yang lebih lebar dan datar.
6. Total luas di bawah kurva adalah 1 (0,5 di sebelah kiri dari rata-rata dan 0,5 di sebelah kanan).
7. Probabilitas untuk variabel acak normal diberikan oleh luas di bawah kurva.

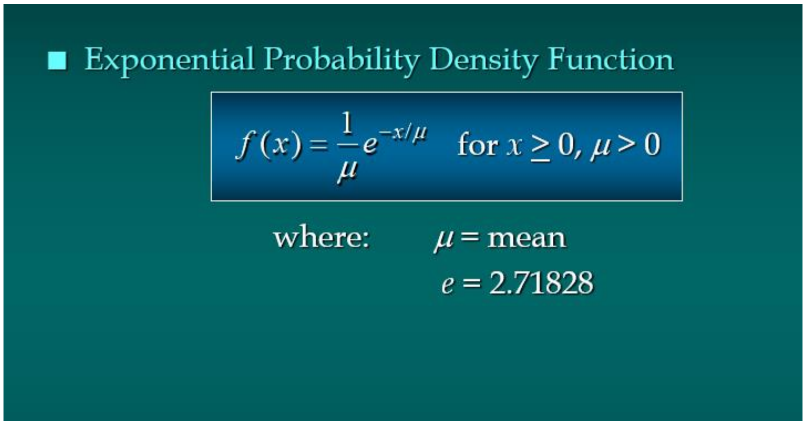
Persentase Nilai dalam Interval yang Umum Digunakan
- 68,26% dari nilai variabel acak normal berada dalam +/- 1 standar deviasi dari rata-ratanya.
- 95,44% dari nilai variabel acak normal berada dalam +/- 2 standar deviasi dari rata-ratanya.
- 99,72% dari nilai variabel acak normal berada dalam +/- 3 standar deviasi dari rata-ratanya.



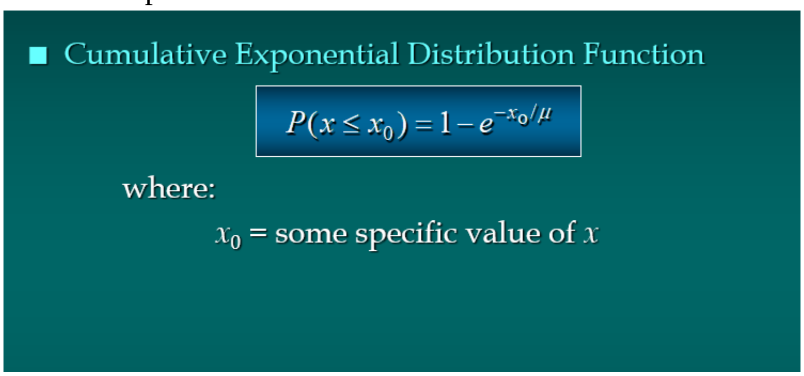
Eksponensial
Eksponensial adalah salah satu distribusi probabilitas yang digunakan untuk menggambarkan waktu yang dibutuhkan untuk terjadi suatu peristiwa tertentu. Misalnya, berapa lama kita harus menunggu sebelum lampu lalu lintas berubah, atau berapa lama sebuah mesin akan rusak. Distribusi eksponensial sering digunakan dalam analisis waktu tunggu dan proses peristiwa yang tidak terduga.

Dalam distribusi eksponensial, kita biasanya tertarik pada dua hal: waktu rata-rata antara peristiwa (biasanya disebut sebagai parameter λ, atau laju kejadian), dan probabilitas bahwa suatu peristiwa terjadi dalam interval waktu tertentu.

Rumus eksponensial:



Rumus eksponensial kumulatif:



Implementasi dalam Python

Anda dapat menggunakan berbagai paket perangkat lunak dalam Python, seperti **scipy.stats**, untuk menghitung probabilitas dari distribusi binomial dan Poisson.

2. Alat dan Bahan

Hardware : Laptop/PC Software: Jupyter Notebook

3. Elemen Kompetensi

a. Latihan pertama – Distribusi Binomial

1. Buka note baru pada Jupyter Notebook
2. Implementasi manual rumus distribusi binomial

Mohammad Afif R. lingkeh
065002400002

Latihan

1. Tinggi badan pemain dalam suatu pertandingan bola basket mengikuti distribusi normal dengan rata-rata tinggi adalah 184 cm. Jika standar deviasi 5 cm.

   a. Tentukan berapa persen kemungkinan tinggi kurang dari 189 cm

   R

   ```
   > pnorm(189,184,5)
   ```

   Output:

   ```
   R 4.4.3 · ~/
   > pnorm(189,184,5)
   [1] 0.8413447
   >
   ```

   Python

   ```python
   from scipy.stats import norm

   # Menentukan rata-rata dan standar deviasi
   mean = 184  # rata-rata
   std_dev = 5  # standar deviasi

   # Menghitung probabilitas tinggi kurang dari 189 cm
   height = 189
   prob_less_than_189 = norm.cdf(height, mean, std_dev)

   # Menampilkan hasil
   print("Probabilitas tinggi kurang dari 189 cm:", prob_less_than_189)
   ```

   ```
   Probabilitas tinggi kurang dari 189 cm: 0.8413447460685429
   ```

   Output:

   ```python
   from scipy.stats import norm
   mean = 184
   sdt_dev = 5

   height = 189
   probability = norm.cdf(height, mean, sdt_dev)
   print('Probabilitas tinggi kurang dari 189 cm : ',probability)
   ```

   ```
   Probabilitas tinggi kurang dari 189 cm :  0.8413447460685429
   ```

   b. Tentukan berapa persen kemungkinan antara 174 cm dan 199 cm

R

> pnorm(199,184,5)-pnorm(174,184,5)

Output:

```
> pnorm(199,184,5)-pnorm(174,184,5)
[1] 0.9759
>
```

Python:

```python
from scipy.stats import norm

# Menentukan rata-rata dan standar deviasi
mean = 184  # rata-rata
std_dev = 5  # standar deviasi

# Menghitung probabilitas tinggi kurang dari 174 cm
prob_less_than_174 = norm.cdf(174, mean, std_dev)

# Menghitung probabilitas tinggi kurang dari 199 cm
prob_less_than_199 = norm.cdf(199, mean, std_dev)

# Kemungkinan antara 174 cm dan 199 cm adalah selisih antara probabilitas kurang dari 199 dan kurang dari 174
prob_between_174_and_199 = prob_less_than_199 - prob_less_than_174

# Menampilkan hasil
print("Probabilitas tinggi antara 174 cm dan 199 cm:", prob_between_174_and_199)
```

```
Probabilitas tinggi antara 174 cm dan 199 cm: 0.9758999700201907
```

Output:

```python
mean = 184
std_dev = 5

prob_less_than_174 = norm.cdf(174, mean, std_dev)
prob_less_than_199 = norm.cdf(199, mean, std_dev)

prob_between_174_and_199 = prob_less_than_199 - prob_less_than_174

print('Probabilitas tinggi antara 174 dan 199 : ', prob_between_174_and_199)
```

```
Probabilitas tinggi antara 174 dan 199 :  0.9758999700201907
```

2. Seorang petani menjual apel ke PT. XYZ. Tim dari bagian pengendalian mutu di PT. XYZ menyatakan bahwa berat apel mengikuti distribusi normal dengan rata-rata 173

gram dan standar deviasi 34 gram. Apel dengan berat kurang dari 120 gram adalah apel yang berkualitas “not good”
Tentukan peluang apel yang berkualitas “not good”

R
> pnorm(120, 173, 34)

Output:

```
> pnorm(120, 173, 34)
[1] 0.05951908
>
```

Python

```python
from scipy.stats import norm

# Menentukan rata-rata dan standar deviasi
mean = 173  # rata-rata
std_dev = 34  # standar deviasi

# Menghitung probabilitas apel berkualitas "not good" (berat kurang dari 120 gram)
prob_not_good = norm.cdf(120, mean, std_dev)

# Menampilkan hasil
print("Peluang apel yang berkualitas 'not good': {:.4f}".format(prob_not_good))
```

```
Peluang apel yang berkualitas 'not good': 0.0595
```

Output:

```
[4] mean = 173
    std_dev = 34

    prob_not_good = norm.cdf(120, mean, std_dev)

    print("Peluang apel yang berkualitas 'not good' : {:.4f}".format(prob_not_good))

    Peluang apel yang berkualitas 'not good' : 0.0595
```

3. Laju kedatangan pelanggan di sebuah apotik berdistribusi Poisson dengan laju rata-rata tiap 10 menit per pelanggan. Waktu pelayanan berdistribusi eksponensial dengan rerata 8 menit tiap layanan.

   Tentukanlah rata-rata jumlah pelanggan yang berada dalam sistem (L) dan rata-rata waktu setiap pelanggan berada dalam sistem (W).

   Python

```python
# Definisi laju kedatangan dan laju pelayanan
arrival_rate = 1/10  # pelanggan per menit
service_rate = 1/8   # pelanggan per menit

# Menghitung rata-rata jumlah pelanggan dalam sistem (L)
L = arrival_rate / (service_rate - arrival_rate)

# Menghitung rata-rata waktu setiap pelanggan berada dalam sistem (W)
W = L / arrival_rate

# Menampilkan hasil
print("Rata-rata jumlah pelanggan dalam sistem (L): {:.2f}".format(L))
print("Rata-rata waktu setiap pelanggan berada dalam sistem (W): {:.2f}".format(W))
```

```
Rata-rata jumlah pelanggan dalam sistem (L): 4.00
Rata-rata waktu setiap pelanggan berada dalam sistem (W): 40.00
```

Output:

```python
[5] # Definisi laju kedatangan dan laju pelayanan
    arrival_rate = 1/10  # pelanggan per menit
    service_rate = 1/8  # pelanggan per menit

    # Menghitung rata-rata jumlah pelanggan dalam sistem (L)
    L = arrival_rate / (service_rate - arrival_rate)

    # Menghitung rata-rata waktu setiap pelanggan berada dalam sistem (W)
    W = L / arrival_rate

    # Menampilkan hasil
    print("Rata-rata jumlah pelanggan dalam sistem (L): {:.2f}".format(L))
    print("Rata-rata waktu setiap pelanggan berada dalam sistem (W): {:.2f}".format(W))

    Rata-rata jumlah pelanggan dalam sistem (L): 4.00
    Rata-rata waktu setiap pelanggan berada dalam sistem (W): 40.00
```

4. Andaikan rerata waktu pelayanan di sebuah kasir supermarket adalah 7 menit dan data diketahui berdistribusi eksponensial.
   Hitunglah peluang seseorang dilayani antara 3 sampai 5 menit

   R

   ```r
   > pexp(5,1/7)-pexp(3,1/7)
   ```

   Output:

   ```
   > pexp(5,1/7)-pexp(3,1/7)
   [1] 0.1618974
   >
   ```

Python

```
from scipy.stats import expon

# Rerata waktu pelayanan (mu)
mean_service_time = 7  # menit

# Menghitung peluang seseorang dilayani antara 3 sampai 5 menit
prob_between_3_and_5 = expon.cdf(5, scale=mean_service_time) - expon.cdf(3, scale=mean_service_time)

# Menampilkan hasil
print("Peluang seseorang dilayani antara 3 sampai 5 menit: {:.7f}".format(prob_between_3_and_5))
```

```
Peluang seseorang dilayani antara 3 sampai 5 menit: 0.1618974
```

Output:

```
from scipy.stats import expon

# Rata-rata waktu pelayanan (mu)
mean_service_time = 7  # menit

# Menghitung peluang seseorang dilayani antara 3 sampai 5 menit
prob_between_3_and_5 = expon.cdf(5, scale=mean_service_time) - expon.cdf(3, scale=mean_service_time)

# Menampilkan hasil
print("Peluang seseorang dilayani antara 3 sampai 5 menit: {:.7f}".format(prob_between_3_and_5))

Peluang seseorang dilayani antara 3 sampai 5 menit: 0.1618974
```

5. Banyaknya pelanggan yang datang per menit pada suatu fasilitas pelayanan penukaran uang untuk lebaran diasumsikan mengikuti distribusi Poisson dengan mean (λ) 5 pelanggan per menit.
Hitunglah peluang bahwa lebih dari 6 pelanggan akan tiba dalam periode 2 menit berikutnya

R

```
> 1-ppois(6,10)
```

Output:

```
> 1-ppois(6,10)
[1] 0.8698586
>
```

Python

```python
from scipy.stats import poisson

# Rata-rata pelanggan per menit (λ)
mean_arrival_rate = 5

# Menghitung peluang lebih dari 6 pelanggan tiba dalam periode 2 menit
prob_more_than_6_arrivals = 1 - poisson.cdf(6, mu=mean_arrival_rate * 2)

# Menampilkan hasil
print("Peluang lebih dari 6 pelanggan akan tiba dalam periode 2 menit berikutnya: {:.7f}".format(prob_more_than_6_arrivals))
```

```
Peluang lebih dari 6 pelanggan akan tiba dalam periode 2 menit berikutnya: 0.8698586
```

Output:

```python
from scipy.stats import poisson

# Rata-rata pelanggan per menit (λ)
mean_arrival_rate = 5

# Menghitung peluang lebih dari 6 pelanggan tiba dalam periode 2 menit
prob_more_than_6_arrivals = 1 - poisson.cdf(6, mu=mean_arrival_rate * 2)

# Menampilkan hasil
print("Peluang lebih dari 6 pelanggan akan tiba dalam periode 2 menit berikutnya: {:.7f}".format(prob_more_than_6_arrivals))
```

```
Peluang lebih dari 6 pelanggan akan tiba dalam periode 2 menit berikutnya: 0.8698586
```

TUGAS

1. Tinggi badan pemain dalam suatu pertandingan bola basket mengikuti distribusi normal dengan rata-rata tinggi adalah 184 cm. Jika standar deviasi 5 cm. Tentukan nilai k sedemikian hingga 16,85% pemain basket lebih tinggi dari k cm

   R

   ```r
   > 1-0.1685
   > qnorm(0.8315,184,5)
   > pnorm(188.8,184,5)
   ```

   Output:

   ```r
   > 1-0.1685
   [1] 0.8315
   > qnorm(0.8315,184,5)
   [1] 188.8005
   > pnorm(188.8,184,5)
   [1] 0.8314724
   >
   ```

Python

```
from scipy.stats import norm

# Menentukan rata-rata dan standar deviasi
mean = 184  # rata-rata
std_dev = 5  # standar deviasi

# Probabilitas tinggi lebih tinggi dari tinggi tertentu
prob_more_than = 1 - 0.1685

# Menggunakan fungsi quantile untuk menemukan tinggi yang sesuai
height = norm.ppf(prob_more_than, mean, std_dev)

# Menampilkan hasil
print("Tinggi sedemikian rupa sehingga 16,85% pemain basket lebih tinggi dari {:.2f} cm".format(height))
```

```
Tinggi sedemikian rupa sehingga 16,85% pemain basket lebih tinggi dari 188.80 cm
```

Output:

```
[8] from scipy.stats import norm

    # Menentukan rata-rata dan standar deviasi
    mean = 184  # rata-rata dalam cm
    std_dev = 5  # standar deviasi dalam cm

    # Probabilitas tinggi lebih tinggi dari tinggi tertentu (1 - 0.1685 = 0.8315)
    prob_more_than = 1 - 0.1685

    # Menggunakan fungsi quantile untuk menemukan tinggi yang sesuai
    height = norm.ppf(prob_more_than, loc=mean, scale=std_dev)

    # Menampilkan hasil
    print("Tinggi sedemikian rupa sehingga 16,85% pemain basket lebih tinggi dari {:.2f} cm".format(height))

    Tinggi sedemikian rupa sehingga 16,85% pemain basket lebih tinggi dari 188.80 cm
```

2. Seorang petani menjual apel ke PT. XYZ. Tim dari bagian pengendalian mutu di PT. XYZ menyatakan bahwa berat apel mengikuti distribusi normal dengan rata-rata 173 gram dan standar deviasi 34 gram. Apel dengan berat kurang dari 120 gram adalah apel yang berkualitas “not good”. Pada sebuah keranjang terdapat 8 apel dari PT. XYZ, tentukan kurang dari sama dengan 2 apel berkualitas “not good”

R

```
pbinom(2,8,0.06)
```

Output:

```
> pbinom(2,8,0.06)
[1] 0.9903771
>
```

Python

```python
from scipy.stats import binom

# Menentukan parameter distribusi binomial
n = 8  # total jumlah apel
p = 0.06  # probabilitas apel berkualitas "not good"

# Menghitung probabilitas jumlah apel berkualitas "not good" kurang dari atau sama dengan 2
prob_not_good_2_or_less = binom.cdf(2, n, p)

# Menampilkan hasil
print("Probabilitas kurang dari atau sama dengan 2 apel berkualitas 'not good': {:.7f}".format(prob_not_good_2_or_less))
```

```
Probabilitas kurang dari atau sama dengan 2 apel berkualitas 'not good': 0.9903771
```

Output:

```python
from scipy.stats import binom

# Menentukan parameter distribusi binomial
n = 8  # total jumlah apel
p = 0.06  # probabilitas apel berkualitas "not good"

# Menghitung probabilitas jumlah apel berkualitas "not good" ≤ 2
prob_not_good_2_or_less = binom.cdf(2, n, p)

# Menampilkan hasil
print("Probabilitas kurang dari atau sama dengan 2 apel berkualitas 'not good': {:.7f}".format(prob_not_good_2_or_less))
```

```
Probabilitas kurang dari atau sama dengan 2 apel berkualitas 'not good': 0.9903771
```

3. Laju kedatangan pelanggan di sebuah apotik berdistribusi Poisson dengan laju rata-rata tiap 10 menit per pelanggan. Waktu pelayanan berdistribusi eksponensial dengan rerata 8 menit tiap layanan. Jika kemudian laju kedatangan pelanggan menurun menjadi 16 menit per pelanggan, berapakah nilai L dan W nya.

   Python

```python
# Definisi laju kedatangan dan laju pelayanan
arrival_rate_new = 1/16  # pelanggan per menit

# Menghitung ulang rata-rata jumlah pelanggan dalam sistem (L)
L_new = arrival_rate_new / (service_rate - arrival_rate_new)

# Menghitung ulang rata-rata waktu setiap pelanggan berada dalam sistem (W)
W_new = L_new / arrival_rate_new

# Menampilkan hasil
print("Rata-rata jumlah pelanggan dalam sistem (L) setelah perubahan: {:.2f}".format(L_new))
print("Rata-rata waktu setiap pelanggan berada dalam sistem (W) setelah perubahan: {:.2f}".format(W_new))
```

```
Rata-rata jumlah pelanggan dalam sistem (L) setelah perubahan: 1.00
Rata-rata waktu setiap pelanggan berada dalam sistem (W) setelah perubahan: 16.00
```

   Output:

```
[10] # Definisi laju kedatangan dan laju pelayanan
     arrival_rate_new = 1/16  # pelanggan per menit (1 pelanggan setiap 16 menit)
     service_rate = 1/8       # pelanggan per menit (1 pelanggan setiap 8 menit)

     # Menghitung ulang rata-rata jumlah pelanggan dalam sistem (L)
     L_new = arrival_rate_new / (service_rate - arrival_rate_new)

     # Menghitung ulang rata-rata waktu setiap pelanggan berada dalam sistem (W)
     W_new = L_new / arrival_rate_new

     # Menampilkan hasil
     print("Rata-rata jumlah pelanggan dalam sistem (L) setelah perubahan: {:.2f}".format(L_new))
     print("Rata-rata waktu setiap pelanggan berada dalam sistem (W) setelah perubahan: {:.2f}".format(W_new))

     Rata-rata jumlah pelanggan dalam sistem (L) setelah perubahan: 1.00
     Rata-rata waktu setiap pelanggan berada dalam sistem (W) setelah perubahan: 16.00
```

4. Andaikan rerata waktu pelayanan di sebuah kasir supermarket adalah 7 menit dan data diketahui berdistribusi eksponensial. Hitung pula peluang seseorang mendapatkan pelayanan lebih dari 8 menit

R

```
> 1-pexp(8,1/7)
```

Output:

```
> 1-pexp(8,1/7)
[1] 0.3189066
>
```

Python

```
from scipy.stats import expon

# Rerata waktu pelayanan (mu)
mean_service_time = 7  # menit

# Menghitung peluang seseorang mendapatkan pelayanan lebih dari 8 menit
prob_more_than_8_minutes = 1 - expon.cdf(8, scale=mean_service_time)

# Menampilkan hasil
print("Peluang seseorang mendapatkan pelayanan lebih dari 8 menit: {:.7f}".format(prob_more_than_8_minutes))
```

```
Peluang seseorang mendapatkan pelayanan lebih dari 8 menit: 0.3189066
```

Output:

```
from scipy.stats import expon

# Rata-rata waktu pelayanan (mu)
mean_service_time = 7  # menit

# Menghitung peluang waktu pelayanan > 8 menit
prob_more_than_8_minutes = 1 - expon.cdf(8, scale=mean_service_time)

# Menampilkan hasil
print("Peluang seseorang mendapatkan pelayanan lebih dari 8 menit: {:.7f}".format(prob_more_than_8_minutes))

Peluang seseorang mendapatkan pelayanan lebih dari 8 menit: 0.3189066
```

Mohammad Afif R. lingkeh
065002400002

5. Banyaknya pelanggan yang datang per menit pada suatu fasilitas pelayanan penukaran uang untuk lebaran diasumsikan mengikuti distribusi Poisson dengan mean (λ) 5 pelanggan per menit. Berapa rata-rata jumlah kedatangan selama periode 1 jam?

   a. Hitunglah peluang bahwa dalam 1 menit berikutnya terdapat tepat 4 pelanggan yang akan datang?

      R

      ```
      > dpois(4,5)
      ```

      Output:

      ```
      > dpois(4,5)
      [1] 0.1754674
      >
      ```

      Python

      ```python
      from scipy.stats import poisson

      # Rata-rata pelanggan per menit (λ)
      mean_arrival_rate = 5

      # Menghitung peluang tepat 4 pelanggan akan datang dalam 1 menit
      prob_4_arrivals = poisson.pmf(4, mu=mean_arrival_rate)

      # Menampilkan hasil
      print("Peluang bahwa dalam 1 menit berikutnya terdapat tepat 4 pelanggan yang akan datang: {:.7f}".format(prob_4_arrivals))
      ```

      ```
      Peluang bahwa dalam 1 menit berikutnya terdapat tepat 4 pelanggan yang akan datang: 0.1754674
      ```

      Output:

      ```
      [12] from scipy.stats import poisson

           # Rata-rata pelanggan per menit (λ)
           mean_arrival_rate = 5

           # Menghitung peluang tepat 4 pelanggan datang dalam 1 menit
           prob_4_arrivals = poisson.pmf(4, mu=mean_arrival_rate)  # Perbaikan: menggunakan parameter bernama 'mu'

           # Menampilkan hasil
           print("Peluang bahwa dalam 1 menit berikutnya terdapat tepat 4 pelanggan yang akan datang: {:.7f}".format(prob_4_arrivals))

           Peluang bahwa dalam 1 menit berikutnya terdapat tepat 4 pelanggan yang akan datang: 0.1754674
      ```

   b. Berapa rata-rata jumlah kedatangan selama periode 1 jam?

      Python

```
# Rata-rata pelanggan per menit (λ)
mean_arrival_rate = 5

# Menghitung rata-rata jumlah kedatangan selama periode 1 jam
mean_arrivals_per_hour = mean_arrival_rate * 60  # 60 menit dalam satu jam

# Menampilkan hasil
print("Rata-rata jumlah kedatangan selama periode 1 jam: {:.2f}".format(mean_arrivals_per_hour))
```

```
Rata-rata jumlah kedatangan selama periode 1 jam: 300.00
```

Output:

```
[13] # Rata-rata pelanggan per menit (λ)
     mean_arrival_rate = 5

     # Menghitung rata-rata jumlah kedatangan selama periode 1 jam
     mean_arrivals_per_hour = mean_arrival_rate * 60  # Konversi menit ke jam (60 menit)

     # Menampilkan hasil
     print("Rata-rata jumlah kedatangan selama periode 1 jam: {:.2f}".format(mean_arrivals_per_hour))

     Rata-rata jumlah kedatangan selama periode 1 jam: 300.00
```

4. File Praktikum

Github Repository:

https://github.com/Afif-lingkeh/praktikum-probstat

5. Kesimpulan

a. Dalam pengerjaan praktikum Statistika, kita dapat mengetahui beberapa contoh penggunaan rumus distribusi normal, poisson, dan exponensial dalam beberapa kasus kehidupan sehari-hari.

b. Kita juga dapat mengetahui berbagai paket perangkat lunak dalam Python, seperti **scipy.stats**, untuk menghitung probabilitas dari distribusi binomial dan Poisson.

6. Cek List (✔)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan	✓	
2.	Tugas	✓	

7. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan	30 Menit	1
2.	Tugas	30 Menit	1

Keterangan:
1. Menarik
2. Baik
3. Cukup
4. Kurang