PEMBAHASAN QUIZ SIMULASI PEMODELAN OKT 2020

Muhammad Reza Fahlevi (NIM : 181401139)

QUIZ SIMULASI PEMODELAN OKT 2020

(KUMPULKAN DALAM FORMAT PDF, MAX SENIN 9 NOV 2020 PKL 20.00)

1. Berat tubuh dalam cm sejumlah mahasiswa dimodelkan sebagai besaran diskrit seperti pada tabel berikut:

Berat (kg)	Frekuensi
60	1
62	4
64	3
66	5
68	4
70	5
72	3
74	2
76	3

a. Berapakah **berat rata-rata** para siswa pria tersebut?

b. Berapakah **kumulatif peluang** untuk berat hingga
   i. 66 kg?
   ii. 70 kg?

c. Tunjukkanlah bahwa **sifat Probability Mass Function** terpenuhi dalam kasus ini!

2. Suatu kantor menggunakan **dua buah PC lama (yang bisa rusak sewaktu-waktu)** untuk mengetik surat dan berkas di kantor tersebut.

- Kalau **kedua PC dapat berjalan** pada pagi hari, maka sorenya ada **30% kemungkinan satu akan rusak dan 15% kemungkinan keduanya rusak.**
- Jika **hanya satu PC yang bisa berjalan** di awal hari, maka ada **35% kemungkinan PC itu akan rusak** di sore hari.
- Jika **semua PC sudah rusak** di pagi hari, kantor terpaksa mengirim semua pekerjaan ke jasa pengetikan di luar kantor.
- PC yang rusak dalam satu hari akan **diambil esok paginya, diperbaiki, dan dipulangkan pagi berikutnya.**
- Sistem diamati **setelah komputer yang diperbaiki dipulangkan** dan **sebelum ada kerusakan baru** lagi.

Berdasarkan suatu **Rantai Markov Waktu Diskrit**, tentukanlah:

a. **State** untuk menggambarkan keadaan di atas (**jelaskan arti masing-masing state** tersebut!)
b. **Diagram transisi** antar **state**
c. **Matriks peluang transisi**
d. Jika hanya ada **seorang teknisi yang butuh 2 hari** untuk memperbaiki sebuah PC, maka tentukanlah **state, diagram transisi, dan matriks peluang** untuk situasi ini!

3. Waktu yang dibutuhkan untuk memperbaiki sepeda motor di sebuah bengkel **terdistribusi secara eksponensial** dengan waktu rata-rata 0.5 jam.

a. Berapakah **peluang** sebuah sepeda motor akan diperbaiki selama **lebih dari 1 jam**?
b. Berapakah **peluang** sebuah sepeda motor **akan diperbaiki selama 2 jam** jika **telah diperbaiki selama 1 jam**?

4. Jumlah nasabah yang hendak bertransaksi di *teller* sebuah bank memiliki **distribusi Poisson** dengan **tingkat kedatangan rata-rata yang konstan** $\lambda = 8$ **orang/jam.**

a. Berapakah peluang akan **tepat ada 11 orang** yang mengantri di *teller* tersebut?
b. Berapa pula peluang nasabah yang mengantri akan berjumlah **lebih dari 6 orang?**

5. Misalkan suatu **ruang ATM** dapat menampung hingga **maksimal 4 orang** nasabah yang **rata-rata waktu antar kedatangannya** adalah **45 detik/nasabah**, dan rata-rata waktu bertransaksi di ATM adalah **90 detik/nasabah.** Tentukanlah:

a. λ dan μ dalam **jlh nasabah/menit**
b. Diagram transisi keadaan
c. Diagram rate
d. Matriks rate
e. $\alpha1, \alpha2, \alpha3, \alpha4$ (Total **rate keluar** dari setiap state)
f. Matriks peluang untuk suatu nilai kecil Δ (ambil suatu nilai Δ tertentu)

ANSWER

Import library untuk visualisasi data

```
library(ggplot2)
```

1. a. Asumsikan kita memiliki data berat tubuh mahasiswa sebagai berikut

```
berat = c(60, 62, 62, 62, 62, 64, 64, 64, 66, 66, 66, 66, 66, 68,
          68, 68, 68, 70, 70, 70, 70, 70, 72, 72, 72, 74, 74, 76,
          76, 76)
print(berat)

##  [1] 60 62 62 62 62 64 64 64 66 66 66 66 66 68 68 68 68 70 70 70 70 70 72
72 72
## [26] 74 74 76 76 76
```

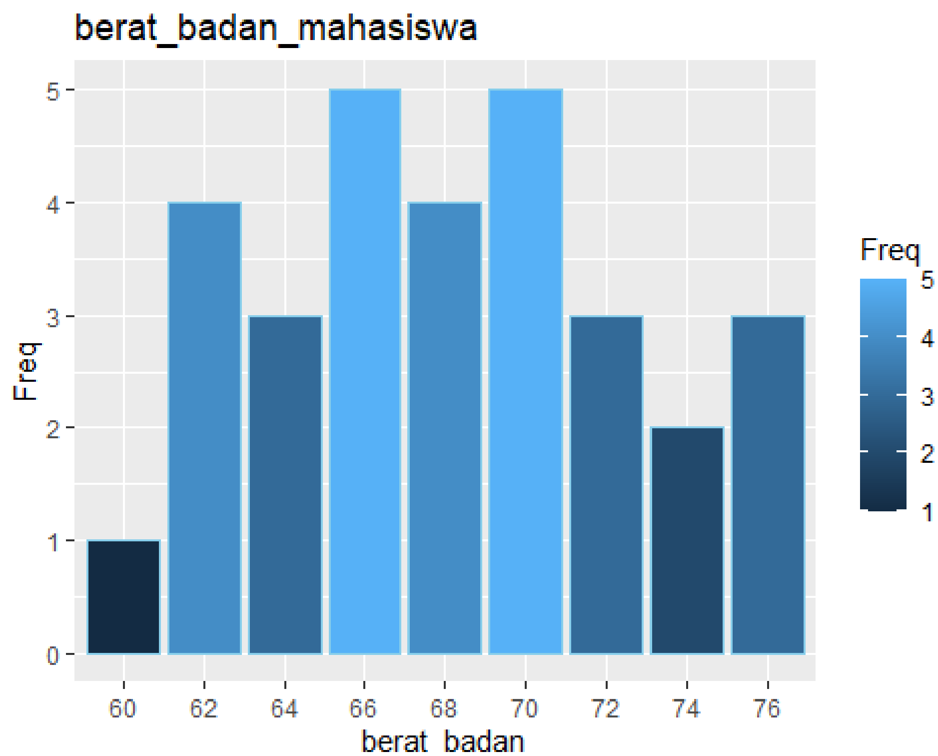
```r
paste("Total datum: ", length(berat))

## [1] "Total datum:  30"
```

Susun data berat tubuh mahasiswa dalam bentuk table dengan kolum berat dan frekuensinya

```r
berat_badan <- factor(berat)
table_berat <- table(berat_badan)
df_berat <- as.data.frame(table_berat)
df_berat

##   berat_badan Freq
## 1          60    1
## 2          62    4
## 3          64    3
## 4          66    5
## 5          68    4
## 6          70    5
## 7          72    3
## 8          74    2
## 9          76    3
```

Data ini dapat divisualisasikan sebagai berikut

```r
plt_berat_badan <- ggplot(data = df_berat, mapping = aes(berat_badan, Freq))
+ geom_bar(stat = "identity", color = "skyblue", aes(fill = Freq)) +
ggtitle("berat_badan_mahasiswa")
plt_berat_badan
```



Berikut adalah deskripsi singkat dari data berat badan mahasiswa

```
summary(berat)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.00   64.50   68.00   68.13   71.50   76.00
```

Karena data berat badan mahasiswa disusun dalam bentuk table dengan kolum berat dan frekuensinya, maka berat badan rata - rata mahasiswa dapat dicari dengan persamaan,

$$\overline{x} = \frac{\sum_{i=1}^{N} f_i \, x_i}{\sum_{i=1}^{N} f \, i}$$

Terlihat bahwa berat badan rata - rata mahasiswa adalah $\overline{x} = 68.13$

b. Asumsikan kita memiliki peubah acak X (X be a random variable) yang mana X adalah banyaknya mahasiswa dengan berat badan $X = x$.

Persamaan (formula) distribusi massa peluang untuk data berat_badan mahasiswa adalah

$$f(X = x) = \frac{f_x}{N}$$

Yang mana f_x adalah frekuensi dari $X = x$ dan N adalah banyaknya datum, yaitu $N = 30$. Karena X adalah peubah acak diskrit, maka persamaan untuk kumulatif peluangnya adalah

$$F(X = x) = \sum_{t \leq x} f\,(t)$$

Komputasi dari persamaan distribusi massa peluang dan kumulatif peluangnya sebagai berikut

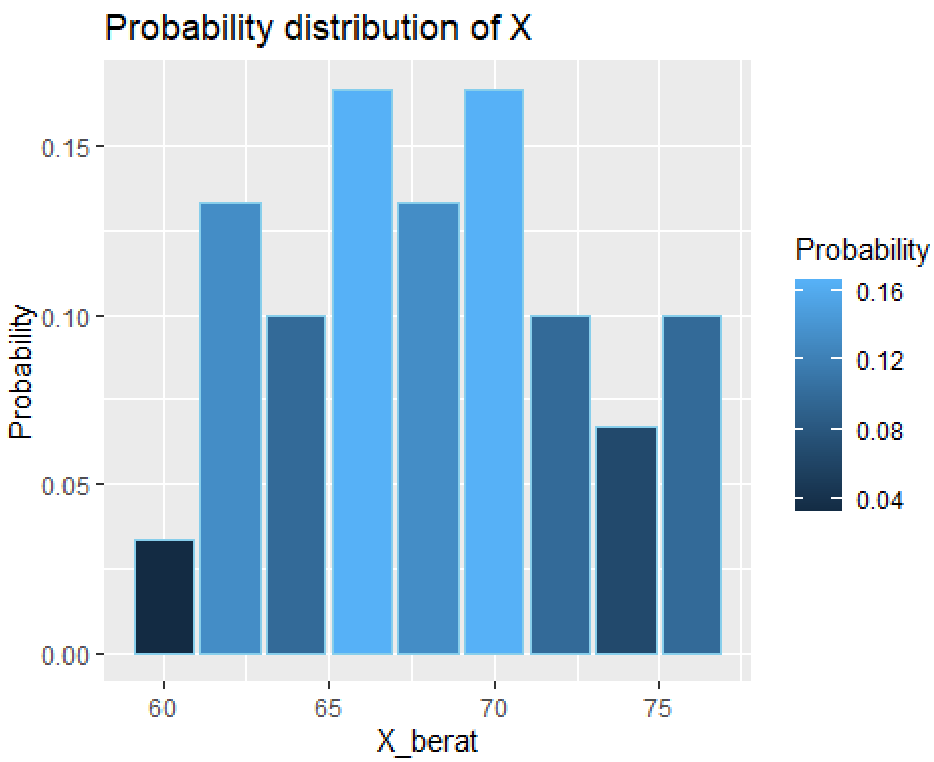
```
#Compute the probability mass function
define_probability <- function(x) {return(df_berat$Freq /
sum(df_berat$Freq))}
df_prob_berat <- data.frame("X_berat" = as.numeric(levels(berat_badan)),
"Probability" = define_probability(as.numeric(levels(berat_badan))))

#Compute the cumulative probability mass function
cumulative_probability <- c()
for (iteration in 1:length(levels(berat_badan))) {
    cumulative_probability[iteration] <-
sum(df_prob_berat$Probability[1:iteration])
}
```

Maka, distribusi massa peluang beserta kumulatif peluang untuk peubah acak X adalah

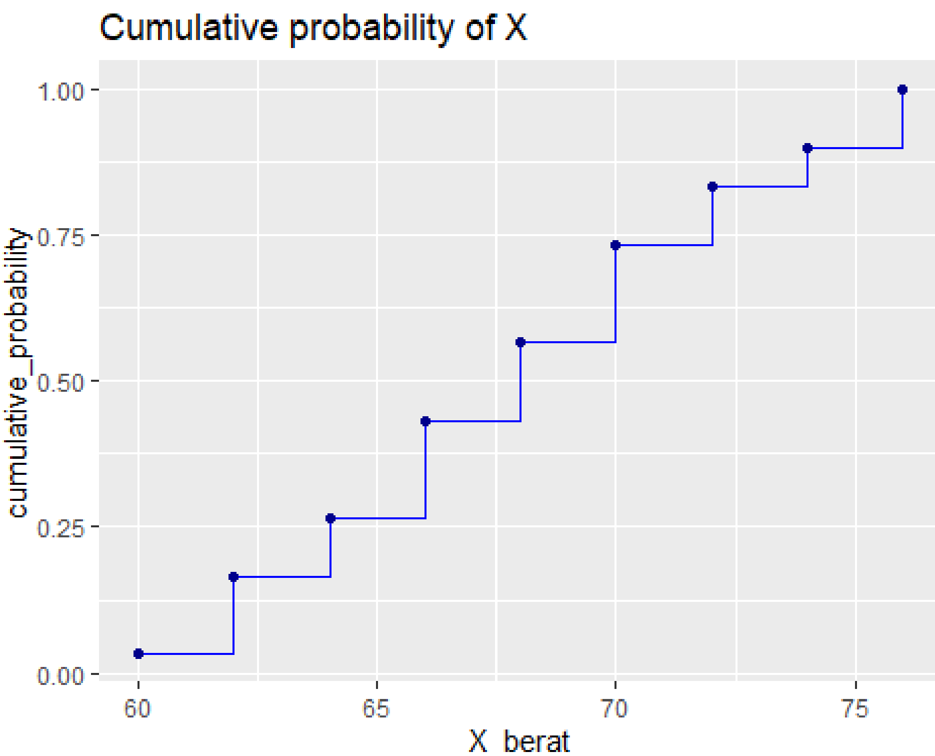
```
#Create data frame that contain X, P(X), and F(X)
df_prob_berat["cumulative.probability"] <- cumulative_probability
df_prob_berat

##   X_berat Probability cumulative.probability
## 1      60  0.03333333             0.03333333
## 2      62  0.13333333             0.16666667
## 3      64  0.10000000             0.26666667
## 4      66  0.16666667             0.43333333
## 5      68  0.13333333             0.56666667
## 6      70  0.16666667             0.73333333
## 7      72  0.10000000             0.83333333
## 8      74  0.06666667             0.90000000
## 9      76  0.10000000             1.00000000
```

Agar makna / informasi dari distribusi peluang ini mudah untuk dipahami, data frame ini dapat divisualisasikan sebagai berikut.

```
plt_prob_berat <- ggplot(data = df_prob_berat, mapping = aes(X_berat,
Probability)) + geom_bar(stat = "identity", color = "skyblue", aes(fill =
Probability)) + ggtitle("Probability distribution of X")
plt_prob_berat
```




```
plt_cmltv_prob_berat <- ggplot(data = df_prob_berat, mapping = aes(X_berat,
cumulative_probability)) + geom_step(color = "blue") + geom_point(color =
"darkblue") + ggtitle("Cumulative probability of X")
plt_cmltv_prob_berat
```



Terlihat bahwa kumulatif peluang untuk berat hingga 66 kg dan 70 kg adalah $F(X = 66) = 0.43 \text{dan} F(X = 70) = 0.73$

c. Untuk membuktikan bahwa sifat **probability mass function** terpenuhi dalam kasus ini, maka untuk kasus ini, kita harus membuktikan bahwa kasus ini memenuhi syarat - syarat distribusi peluang.

Himpunan dari pasangan berurut $(x, f(x))$ **adalah suatu fungsi peluang, fungsi massa peluang, atau distribusi peluang (probability function, probability mass function, or probability distribution) dari peubah acak** X **jika, untuk kemungkinan nilai x,**

1. $f(x) \geq 0$,
2. $\sum_x f(x) = 1$,
3. $P(X = x) = f(x)$.

Untuk kasus ini, kita memiliki bahwa untuk $f(X = x) = f_x/N$, nilai $f(x) \geq 0$. Artinya, untuk semua kemungkinan nilai x, maka nilai $f(x)$ tidak bernilai negatif. Dengan demikian, sifat pertama probability mass function terpenuhi untuk kasus ini.

Jumlah dari peluang untuk nilai x yang mungkin untuk kasus ini adalah,

$$
\begin{aligned}
\sum_{\forall x} f\,(x) &= f(60) + f(62) + f(64) + f(66) + f(68) + f(70) + f(72) + f(74) + f(76) \\
&= \frac{1}{30} + \frac{4}{30} + \frac{3}{30} + \frac{5}{30} + \frac{4}{30} + \frac{5}{30} + \frac{3}{30} + \frac{2}{30} + \frac{3}{30} \\
&= \frac{30}{30} \\
\therefore \sum_{\forall x} f\,(x) &= 1
\end{aligned}
$$

Atau

```
sum(df_prob_berat$Probability)

## [1] 1
```

Karena jumlah $f\,(x)$ untuk semua kemungkinan nilai x adalah 1. Dengan demikan, kasus ini memenuhi sifat kedua probability mass function.

Persamaan atau formula untuk $P(X = x)$ untuk kasus ini didefinisikan sebagai,

$$f\,(x) = \frac{f_x}{N}$$

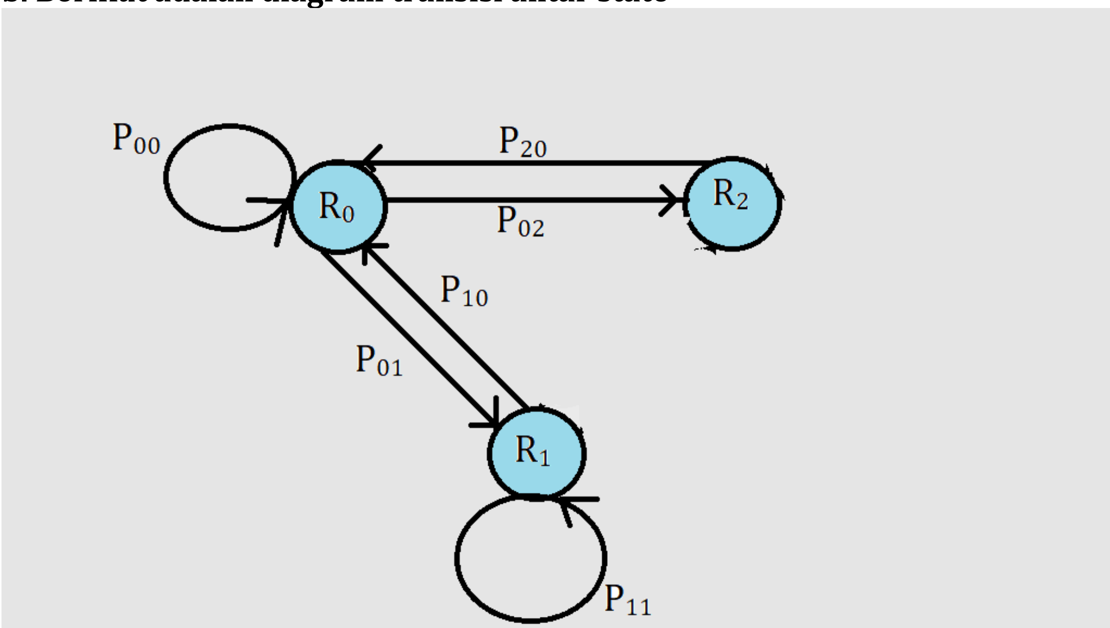
dengan f_x adalah frekuensi dari nilai x. Maka, $P(X = x)$ terdefinisi. Dengan demikian, kasus ini memenuhi sifat ketiga dari probability mass function.

Ketika telah ditunjukkan bahwa kasus ini memenuhi ketiga sifat dari probability mass function. Dengan demikian, telah terbukti bahwa probability mass function untuk kasus ini adalah sah (valid).

2. a. Karena ada dua buah PC (yang bisa rusak sewaktu-waktu), maka asumsikan R_i adalah peubah acak stokastik dan R_i adalah banyaknya komputer yang rusak, dan $i = 0,1,2$. Misalkan, R_i menyatakan bahwa terdapat i komputer yang rusak.

State untuk menggambarkan keadaan di atas adalah $S = \{R_0, R_1, R_2\}$

b. Berikut adalah diagram transisi antar state



c. Perlu diketahui bahwa untuk sifat Markov, maka setiap R_i independen dengan yang lainnya. Maka $P(R_i|R_j) = P(R_i)$, $P(R_i \cup R_j) = P(R_i) + P(R_j)$, dan $P(R_i \cap R_j) = P(R_i)P(R_j)$.

$$P_{3\times3} = \begin{bmatrix} P(R_0|R_0) & P(R_1|R_0) & P(R_2|R_0) \\ P(R_0|R_1) & P(R_1|R_1) & P(R_2|R_1) \\ P(R_0|R_2) & P(R_1|R_2) & P(R_2|R_2) \end{bmatrix}$$

Perlu diingat bahwa, *sistem diamati setelah komputer yang diperbaiki dipulangkan dan sebelum ada kerusakan baru lagi.* Jadi, $P(R_i|R_j)$ berarti R_j telah diperbaiki dan dipulangkan, sehingga R_j tidak mungkin rusak pada hari itu juga R_j dipulangkan. Dan pada hari R_j dipulangkan, berapakah peluang R_i.

- Kalau kedua PC dapat berjalan pada pagi hari, maka sorenya ada 30% kemungkinan satu akan rusak dan 15% kemungkinan keduanya rusak. Berarti $P(R_1|R_0) = 0.3$ dan $P(R_2|R_0) = 0.15$. Karena pada rantai markov jumlah baris pada matrix peluang transisinya sama dengan 1. Jadi,

$$\begin{aligned} P(R_0|R_0) &= 1 - [P(R_1|R_0) + P(R_2|R_0)] \\ &= 1 - [0.30 + 0.15] = 1 - 0.45 \\ &= 0.55 \end{aligned}$$

- Jika hanya satu PC yang bisa berjalan di awal hari, maka ada 35% kemungkinan PC itu akan rusak di sore hari. Maka, $P(R_1|R_1) = 0.35$ dan $P(R_2|R_1) = 0$, $P(R_2|R_1) = 0$ dikarenakan pada saat salah satu PC yang diperbaiki dan dipulangkan pada hari itu juga tidak boleh rusak, sehingga tidak mungkin pada hari tersebut kedua PC rusak. Karena pada rantai markov jumlah baris pada matrix peluang transisinya sama dengan 1. Jadi,

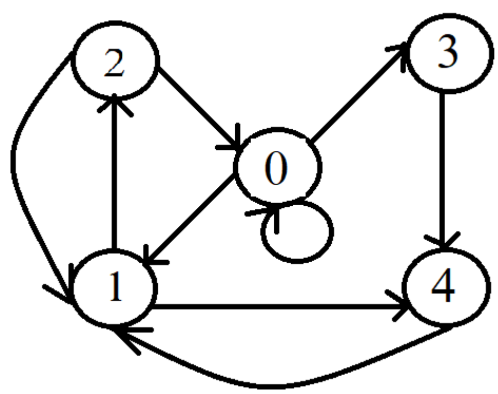
$$\begin{aligned} P(R_0|R_1) &= 1 - [P(R_1|R_1) + P(R_2|R_1)] \\ &= 1 - [0.35 + 0] = 1 - 0.35 \\ &= 0.65 \end{aligned}$$

- PC yang rusak dalam satu hari akan diambil esok paginya, diperbaiki, dan dipulangkan pagi berikutnya. Maka, $P(R_0|R_2) = 1$. Dan lagi, Karena pada rantai markov jumlah baris pada matrix peluang transisinya sama dengan 1. Jadi, $P(R_1|R_2) = 0$ dan $P(R_2|R_2) = 0$.

Dengan demikian, matrix peluang transisinya adalah

$$P_{3\times 3} = \begin{bmatrix} 0.55 & 0.30 & 0.15 \\ 0.65 & 0.35 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

d. Statenya adalah S = {s_1, s_2}, dengan s_1 adalah hari perbaikan komputer pertama dan s_2 adalah status komputer kedua (rusak atau tidak). Diagram transisi statenya



adalah

Matrix peluang untuk situasi ini adalah

$$P_{5\times 5} = \begin{bmatrix} 0.55 & 0.30 & 0 & 0.1 & 0 \\ 0 & 0 & 0.65 & 0 & 0.35 \\ 0.65 & 0.35 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

3. Asumsikan T adalah waktu yang dibutuhkan untuk memperbaiki sepeda motor di sebuah bengkel dan T adalah peubah acak kontinu (continuous random variable). Telah disebutkan bahwa T dimodelkan dengan distribusi eksponensial dengan waktu rata - rata $\beta = 0.5$ jam.

Definisi density function untuk distribusi eksponensial

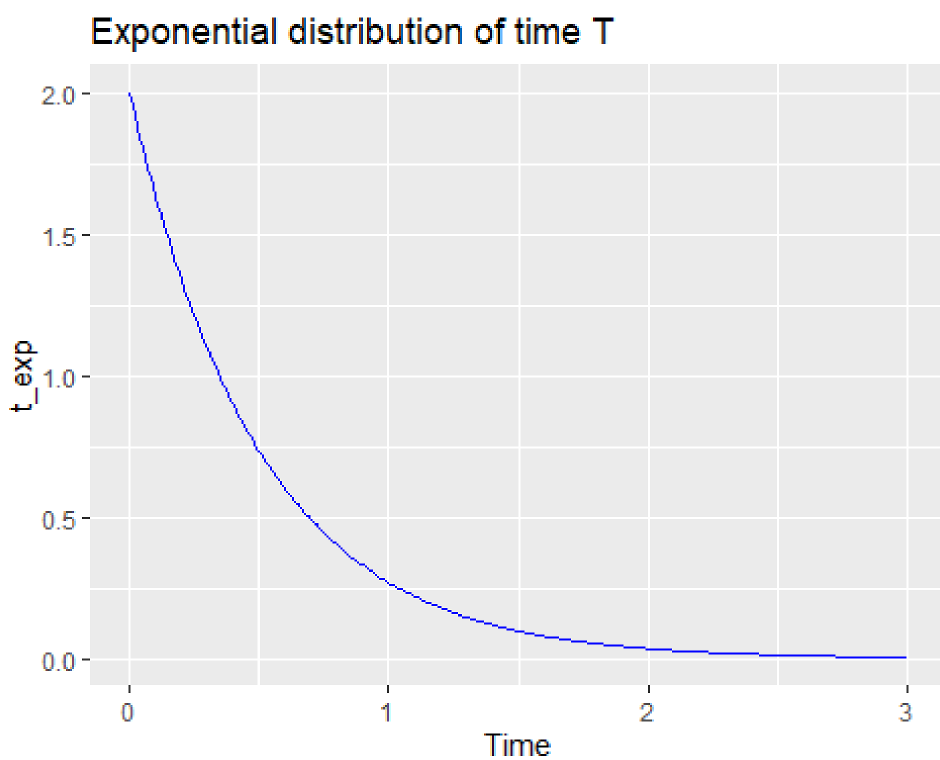
$$f(x;\beta) = \begin{cases} \frac{1}{\beta} e^{-\frac{x}{\beta}}, & \text{if} x > 0 \\ 0 & otherwise \end{cases}$$

Dengan $e = 2.7182818\ldots$ dan $\beta > 0$. Definisikan $T = x$ jam dan $f(x;\beta = 0.5)$

```
T_randvar <- seq(0, 3, 0.01)
define_exponential <- function(randvar, bta) {return((1 / bta) * exp(-randvar
/ bta))}
df_exponential <- data.frame("Time" = T_randvar, "t_exp" =
define_exponential(T_randvar, bta = 0.5))
#print(df_exponential) Look at appendix
```

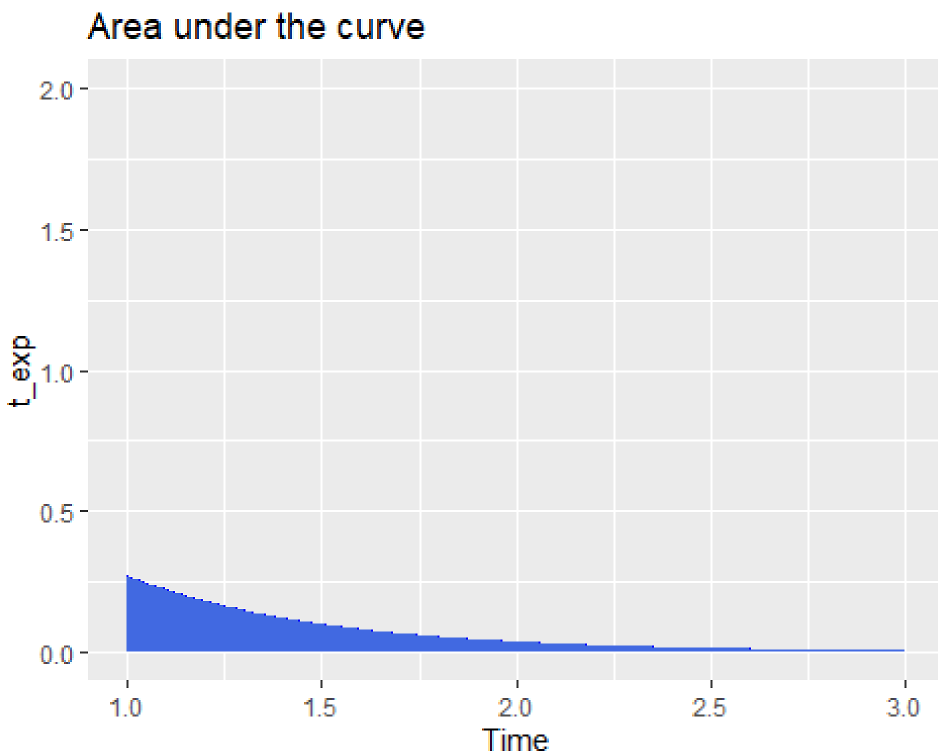
Plot untuk distribusi eksponensial $f(x;\beta = 0.5)$

```
library(ggplot2)
plt_exponential <- ggplot(data = df_exponential, mapping = aes(Time, t_exp))
+ geom_line(color = "blue") + ggtitle("Exponential distribution of time T")
plt_exponential
```



a. Peluang sebuah sepeda motor akan diperbaiki selama lebih dari 1 jam adalah luas area di bawah kurva $f(x; \beta = 0.5)$ untuk x > 1. Yaitu,

```
plt_area_under_the_curve <- plt_exponential + geom_area(fill = "royalblue") +
xlim(1, 3)
plt_area_under_the_curve + ggtitle("Area under the curve")
```



$$
\begin{aligned}
P(T > 1) &= \int_1^{\infty} f(x; \beta = 0.5) dx \\
&= \lim_{t \to \infty} \int_1^t \frac{1}{\beta} e^{-\frac{x}{\beta}} dx \\
&= \frac{1}{\beta} \lim_{t \to \infty} \int_1^t e^{-\frac{x}{\beta}} dx
\end{aligned}
$$

Asumsikan $u = -x/\beta$, maka $du = -dx/\beta$. Maka, untuk $x = 1$, diperoleh $u = -1/\beta$, dan untuk $x = \infty$, $u = -\infty$. Jadi,

$$\begin{aligned} P(T > 1) &= \frac{1}{\beta} \lim_{t \to \infty} \int_1^t e^{-\frac{x}{\beta}}\, dx \\ &= \frac{-\beta}{\beta} \lim_{t \to -\infty} \int_{-1/\beta}^{t} e^u\, du \\ &= -1 \lim_{t \to -\infty} e^u \Big|_{u=-1/\beta}^{u=t} \\ &= -1 \lim_{t \to -\infty} (e^t - e^{-\frac{1}{\beta}}) \\ &= -1(0 - e^{\frac{1}{\beta}}) \\ &= e^{-\frac{1}{\beta}} \end{aligned}$$

Karena $\beta = 0.5$, **maka** $P(T > 1) = e^{-\frac{1}{0.5}}$ **atau** $P(T > 1) \approx 0.1353$.

b. Perlu diingat bahwa pada distribusi eksponensial adalah sifatnya yang tanpa memori (memoryless or lack of memory). Misalkan, peluang sepeda motor akan diperbaiki selama lebih dari t_0 adalah $P(T \geq t_0)$, maka peluang lamanya sepeda motor akan diperbaiki selama lebih dari atau sama dengan $t_0 + t$ adalah,

$$P(T \geq t_0 + t | T \geq t_0) = P(T \geq t_0)$$

Dengan demikian, peluang sebuah sepeda motor akan diperbaiki selama 2 jam jika telah diperbaiki selama 1 jam, adalah sama dengan peluang sepeda motor akan diperbaiki selama lebih dari atau sama dengan 1 jam.

Yaitu,

$$\begin{aligned} P(T \geq t_0 + t | T \geq t_0) &= P(T \geq t_0) \\ P(T \geq 1 + 1 | T \geq 1) &= P(T \geq 1) \\ P(T \geq 2 | T \geq 1) &= P(T \geq 1) \\ \therefore P(T \geq 2 | T \geq 1) &= 0.1353 \end{aligned}$$

4. Asumsikan X adalah jumlah nasabah yang hendak bertransaksi di *teller* sebuah bank dan X adalah peubah acak diskrit (discrete random variable). Telah disebutkan bahwa X dimodelkan sebagai distribusi Poisson dengan tingkat kedatangan rata - rata yang konstan, $\lambda = 8$ orang/jam.

Karena X adalah peubah acak Poisson (Poisson random variable), maka X merupakan proses Poisson (Poisson proccess) dan distribusi peluang X adalah distribusi Poisson. Definisi distribusi Poisson adalah

$$p(x;\lambda t) \quad = \frac{e^{-\lambda t}(\lambda t)^x}{x!}, \quad x = 0,1,2\ldots$$

dengan λ adalah angka rata-rata dari nilai kemungkinan per satuan waktu, jarak, area, atau volume. Sedangkan jumlah peluang Poisson atau peluang kumulatifnya adalah

$$P(r;\lambda t) = \sum_{x=0}^{r} p\,(x;\lambda t) = \sum_{x=0}^{r} \frac{e^{-\lambda t}(\lambda t)^x}{x!}$$

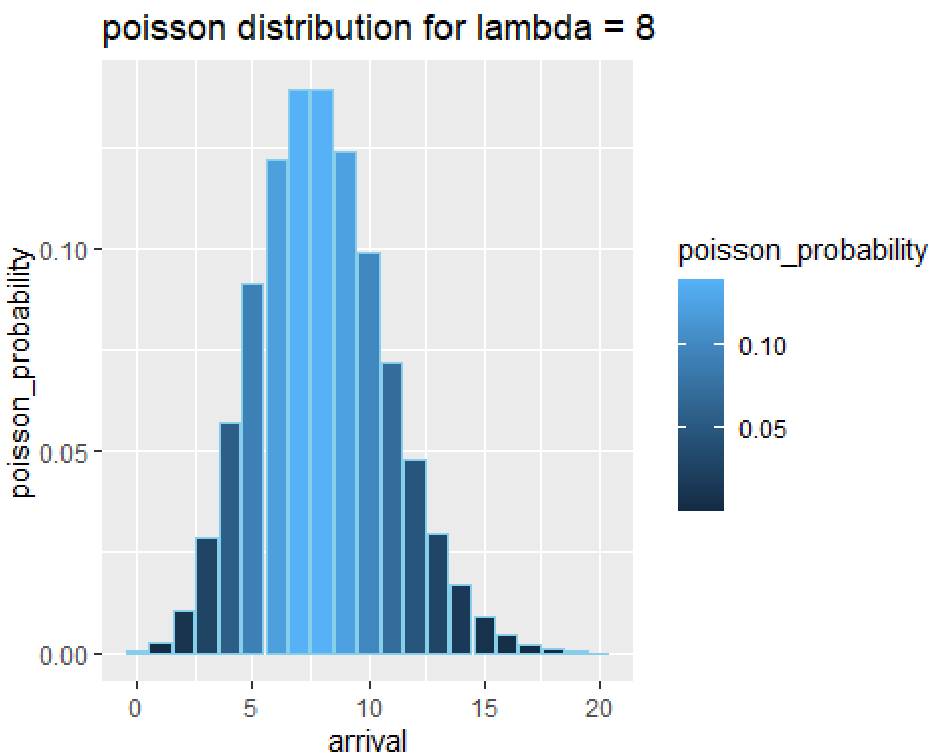
dengan $r \geq 0$.

Definisikan domain X dan distribusi Poisson.

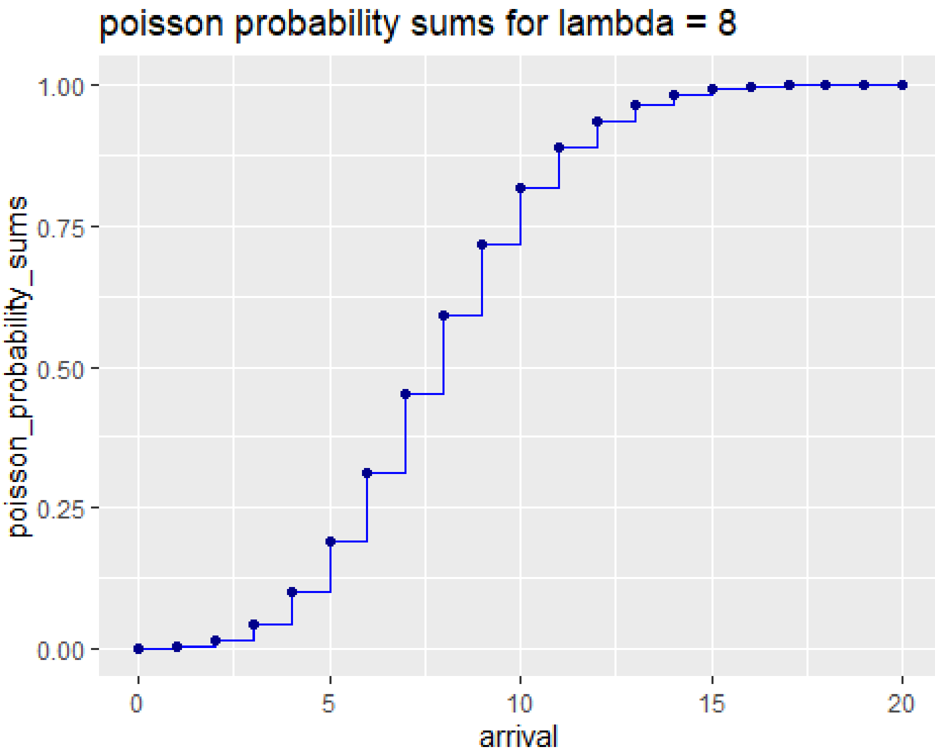
```
#In this case, we let X = 0, 1, ... , 20
#We still being able to compute X > 20 by using define_poisson function
arrival <- seq(0, 20, 1)
define_poisson <- function(X, lambda) {return(exp(-lambda) * (lambda ** X) /
factorial(X))}
define_cmltv_poisson <- function(r, lambda) {return(sum(define_poisson(seq(0,
r, 1), lambda)))}
cmltv_prob_arrival <- c(sum(define_poisson(0:0, lambda = 8)))
for (arrivals in 2:length(arrival)) {
    cmltv_prob_arrival[arrivals] = sum(define_poisson(0:(arrivals-1), lambda
= 8))
}
df_poisson <- data.frame("arrival" = arrival[0:21], "poisson_probability" =
define_poisson(arrival, lambda = 8), "poisson_probability_sums" =
cmltv_prob_arrival[0:21])
#print(df_poisson) Look at appendix
```

Plot untuk distribusi poisson beserta peluang peluang kumulatifnya.

```
library(ggplot2)
plt_poisson <- ggplot(data = df_poisson, mapping = aes(arrival,
poisson_probability)) + geom_bar(stat = "identity", color = "skyblue",
aes(fill = poisson_probability)) + ggtitle("poisson distribution for lambda =
8")
plt_poisson
```




```
plt_poisson_prob_sums <- ggplot(data = df_poisson, mapping = aes(arrival,
poisson_probability_sums)) + geom_step(color = "blue") + geom_point(color =
"darkblue") + ggtitle("poisson probability sums for lambda = 8")
plt_poisson_prob_sums
```



a. Berdasarkan grafik distribusi poisson di atas, dengan nilai $\lambda = 8/jam$ atau $\lambda jam = 8$ atau $\lambda t = 8$. Maka, untuk $X = 11$, diperoleh $P(X = 11) = 0.0723$, dengan perhitungan sebagai berikut

$$p(x = 11; \lambda t = 8) = \frac{e^{-8} \times 8^{11}}{11!} = 0.0723$$

Maka $P(X = 11) = 0.0723$

b. Dengan menggunakan prinsip total peluang untuk semua kemungkinan nilai X adalah sama dengan 1, maka

$$P(X \leq 6) + P(X > 6) = 1$$

Diperoleh

$$\begin{aligned} P(X > 6) &= 1 - P(X \leq 6) = 1 - P(r = 6; \lambda t = 8) \\ &= 1 - \sum_{x=0}^{6} \frac{e^{-8} 8^x}{x!} \\ &= 1 - 0.3134 \\ \therefore P(X > 6) &= 0.6866 \end{aligned}$$

Dengan demikian, peluang nasabah yang mengantri akan berjumlah lebih dari 6 orang adalah $P(X > 6) = 0.6866$

5. Asumsikan X_i adalah lamanya waktu yang diperlukan antar kedatangan para nasabah dan X_j adalah lamanya waktu nasabah bertransaksi di ATM. Maka, X_i dan X_j memiliki distribusi eksponensial sebagai berikut.

$$f(x;\beta_i) = \frac{1}{\beta_i} e^{-\frac{x}{\beta_i}} \qquad dan \qquad f(x;\beta_j) = \frac{1}{\beta_j} e^{-\frac{x}{\beta_j}}$$

a. Jika dinotasikan β_i adalah rata - rata waktu antar kedatangannya para nasabah dan β_j adalah rata-rata waktu bertransaksi di ATM, maka $\beta_i = 45$ detik/nasabah dan $\beta_j = 90$ detik/nasabah atau, $\beta_i = 0.75$ menit/nasabah dan $\beta_j = 1.5$ menit/nasabah. Untuk distribusi eksponensial, nilai ekspektasinya adalah $\mu = 1/\beta$. Jadi,

$$\mu_i = \frac{1}{0.75} \qquad dan \qquad \mu_j = \frac{1}{1.5}$$

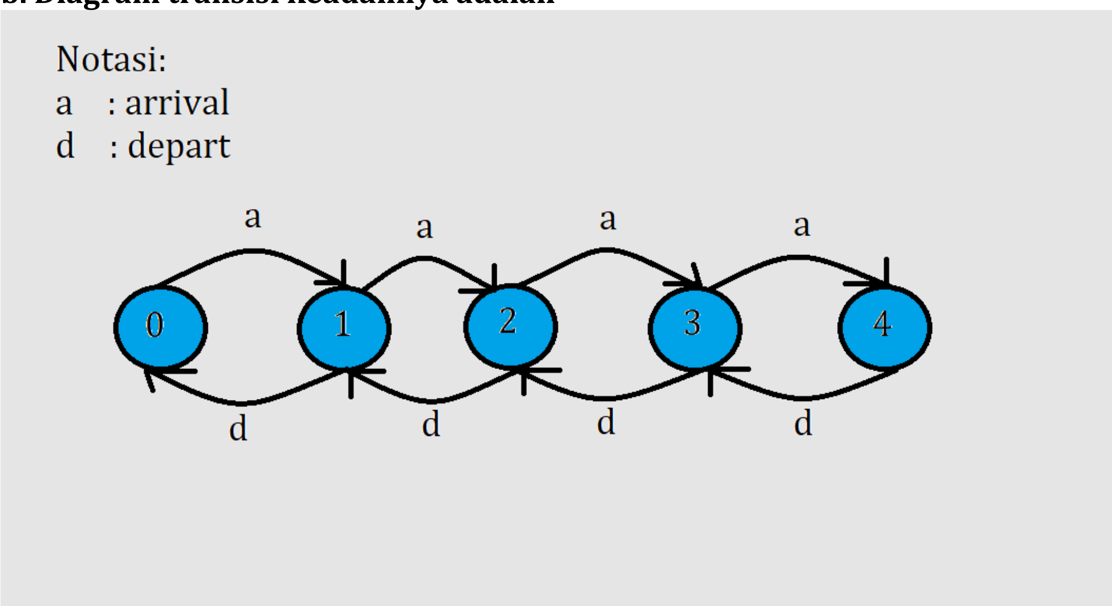
atau

$$\mu_i = 1.3 \qquad dan \qquad \mu_j = 0.6$$

Jika dinotasikan $\lambda = \mu_i$ dan $\mu = \mu_j$, maka $\lambda = 1.3/menit$ dan $\mu = 0.6/menit$.

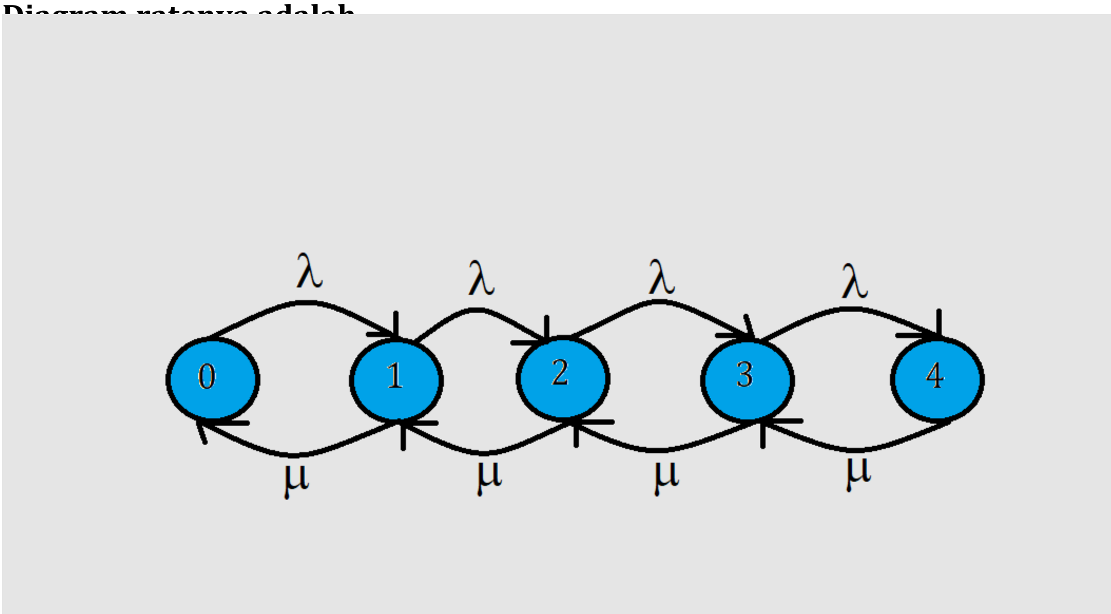
Maka $\lambda = 1.3/menit$ **dan** $\mu = 0.6/menit$

b. Diagram transisi keadannya adalah



c. Diagram rate adalah diagram yang dibuat dari diagram transisi dengan menggantikan simbol aktifitas dengan tingkat (kecepatan, atau yang memiliki besaran *something*/*time*) aktifitasnya.

Diagram ratenya adalah



d. Matrix rate adalah matrix $R_{m \times n} = (r_{ij})$. Berdasarkan diagram ratenya pada point (c), matrix rate $R = (r_{ij})$ adalah

$$R_{5\times5} = \begin{bmatrix} 0 & \lambda & 0 & 0 & 0 \\ \mu & 0 & \lambda & 0 & 0 \\ 0 & \mu & 0 & \lambda & 0 \\ 0 & 0 & \mu & 0 & \lambda \\ 0 & 0 & 0 & \mu & 0 \end{bmatrix}$$

e. Jika α_i adalah total rate keluar dari setiap state, maka α_i dapat dihitung dengan persamaan

$$\alpha_i = \sum_{j=0}^{m-1} r_{ij}$$

maka,

$$\begin{aligned} \alpha_0 &= r_{0.0} + r_{0.1} + r_{0.2} + r_{0.3} + r_{0.4} \\ &= 0 + \lambda + 0 + 0 + 0 \\ \therefore \alpha_0 &= \lambda \end{aligned}$$

Berdasarkan diagram ratenya pada point (c), diperoleh

$$\alpha_i = \lambda + \mu \qquad \text{for} i = 1,2,3$$

dan $\alpha_4 = \mu$.

Dengan demikian, $\alpha_0 = 1.3, \alpha_1 = \alpha_2 = \alpha 3 = 1.9, \text{dan} \alpha_4 = 0.6$

f. Karena jumlah pada baris matrix peluang transisi state sama dengan 1, maka

$$
\mathbf{P} = \begin{bmatrix} 1-\Delta\lambda & \Delta\lambda & 0 & 0 & 0 \\ \Delta\mu & 1-\Delta(\mu+\lambda) & \Delta\lambda & 0 & 0 \\ 0 & \Delta\mu & 1-\Delta(\mu+\lambda) & \Delta\lambda & 0 \\ 0 & 0 & \Delta\mu & 1-\Delta(\mu+\lambda) & \Delta\lambda \\ 0 & 0 & 0 & \Delta\mu & 1-\Delta\mu \end{bmatrix}
$$

Untuk nilai $\lambda = 1.3$ dan $\mu = 0.6$, Maka

$$
\mathbf{P} = \begin{bmatrix} 1-\Delta 1.3 & \Delta 1.3 & 0 & 0 & 0 \\ \Delta 0.6 & 1-\Delta 1.9 & \Delta 1.3 & 0 & 0 \\ 0 & \Delta 0.6 & 1-\Delta 1.9 & \Delta 1.3 & 0 \\ 0 & 0 & \Delta 0.6 & 1-\Delta 1.9 & \Delta 1.3 \\ 0 & 0 & 0 & \Delta 0.6 & 1-\Delta 0.6 \end{bmatrix}
$$

Untuk suatu interval kecil waktu $\Delta = 0.025$, maka

$$
\mathbf{P} = \begin{bmatrix} 1-0.025\times 1.3 & 0.025\times 1.3 & 0 & 0 & 0 \\ 0.025\times 0.6 & 1-0.025\times 1.9 & 0.025\times 1.3 & 0 & 0 \\ 0 & 0.025\times 0.6 & 1-0.025\times 1.9 & 0.025\times 1.3 & 0 \\ 0 & 0 & 0.025\times 0.6 & 1-0.025\times 1.9 & 0.025\times 1.3 \\ 0 & 0 & 0 & 0.025\times 0.6 & 1-0.025\times 0.6 \end{bmatrix}
$$

Dengan demikian, matrix peluang untuk suatu interval kecil waktu $\Delta = 0.025$ adalah

$$
\mathbf{P} = \begin{bmatrix} 0.9675 & 0.0325 & 0 & 0 & 0 \\ 0.015 & 0.9525 & 0.0325 & 0 & 0 \\ 0 & 0.015 & 0.9525 & 0.0325 & 0 \\ 0 & 0 & 0.015 & 0.9525 & 0.0325 \\ 0 & 0 & 0 & 0.015 & 0.985 \end{bmatrix}
$$

APPENDIX

A.1. Nilai $f(x; \beta = 0.5)$

```
T_randvar <- seq(0, 3, 0.01)
define_exponential <- function(randvar, bta) {return((1 / bta) * exp(-randvar
/ bta))}
df_exponential <- data.frame("Time" = T_randvar, "t_exp" =
define_exponential(T_randvar, bta = 0.5))
print(df_exponential)
```

```
##      Time       t_exp
## 1    0.00 2.000000000
## 2    0.01 1.960397347
## 3    0.02 1.921578878
## 4    0.03 1.883529067
## 5    0.04 1.846232693
## 6    0.05 1.809674836
## 7    0.06 1.773840873
## 8    0.07 1.738716471
## 9    0.08 1.704287578
## 10   0.09 1.670540423
```

```
## 11  0.10 1.637461506
## 12  0.11 1.605037596
## 13  0.12 1.573255722
## 14  0.13 1.542103172
## 15  0.14 1.511567483
## 16  0.15 1.481636441
## 17  0.16 1.452298074
## 18  0.17 1.423540646
## 19  0.18 1.395352652
## 20  0.19 1.367722818
## 21  0.20 1.340640092
## 22  0.21 1.314093640
## 23  0.22 1.288072842
## 24  0.23 1.262567291
## 25  0.24 1.237566784
## 26  0.25 1.213061319
## 27  0.26 1.189041096
## 28  0.27 1.165496505
## 29  0.28 1.142418128
## 30  0.29 1.119796733
## 31  0.30 1.097623272
## 32  0.31 1.075888875
## 33  0.32 1.054584848
## 34  0.33 1.033702669
## 35  0.34 1.013233985
## 36  0.35 0.993170608
## 37  0.36 0.973504512
## 38  0.37 0.954227831
## 39  0.38 0.935332854
## 40  0.39 0.916812023
## 41  0.40 0.898657928
## 42  0.41 0.880863309
## 43  0.42 0.863421047
## 44  0.43 0.846324165
## 45  0.44 0.829565823
## 46  0.45 0.813139319
## 47  0.46 0.797038082
## 48  0.47 0.781255671
## 49  0.48 0.765785772
## 50  0.49 0.750622198
## 51  0.50 0.735758882
## 52  0.51 0.721189880
## 53  0.52 0.706909364
## 54  0.53 0.692911621
## 55  0.54 0.679191051
## 56  0.55 0.665742167
## 57  0.56 0.652559589
## 58  0.57 0.639638044
## 59  0.58 0.626972362
## 60  0.59 0.614557477
```

```
## 61  0.60 0.602388424
## 62  0.61 0.590460334
## 63  0.62 0.578768436
## 64  0.63 0.567308053
## 65  0.64 0.556074601
## 66  0.65 0.545063586
## 67  0.66 0.534270604
## 68  0.67 0.523691337
## 69  0.68 0.513321554
## 70  0.69 0.503157106
## 71  0.70 0.493193928
## 72  0.71 0.483428034
## 73  0.72 0.473855517
## 74  0.73 0.464472549
## 75  0.74 0.455275377
## 76  0.75 0.446260320
## 77  0.76 0.437423774
## 78  0.77 0.428762203
## 79  0.78 0.420272142
## 80  0.79 0.411950196
## 81  0.80 0.403793036
## 82  0.81 0.395797398
## 83  0.82 0.387960085
## 84  0.83 0.380277960
## 85  0.84 0.372747952
## 86  0.85 0.365367048
## 87  0.86 0.358132296
## 88  0.87 0.351040801
## 89  0.88 0.344089728
## 90  0.89 0.337276295
## 91  0.90 0.330597776
## 92  0.91 0.324051502
## 93  0.92 0.317634852
## 94  0.93 0.311345261
## 95  0.94 0.305180212
## 96  0.95 0.299137238
## 97  0.96 0.293213924
## 98  0.97 0.287407900
## 99  0.98 0.281716842
## 100 0.99 0.276138475
## 101 1.00 0.270670566
## 102 1.01 0.265310930
## 103 1.02 0.260057422
## 104 1.03 0.254907940
## 105 1.04 0.249860424
## 106 1.05 0.244912857
## 107 1.06 0.240063257
## 108 1.07 0.235309686
## 109 1.08 0.230650242
## 110 1.09 0.226083061
```

```
## 111 1.10 0.221606317
## 112 1.11 0.217218218
## 113 1.12 0.212917009
## 114 1.13 0.208700970
## 115 1.14 0.204568413
## 116 1.15 0.200517687
## 117 1.16 0.196547171
## 118 1.17 0.192655276
## 119 1.18 0.188840446
## 120 1.19 0.185101155
## 121 1.20 0.181435907
## 122 1.21 0.177843235
## 123 1.22 0.174321703
## 124 1.23 0.170869902
## 125 1.24 0.167486451
## 126 1.25 0.164169997
## 127 1.26 0.160919213
## 128 1.27 0.157732800
## 129 1.28 0.154609481
## 130 1.29 0.151548008
## 131 1.30 0.148547156
## 132 1.31 0.145605726
## 133 1.32 0.142722539
## 134 1.33 0.139896443
## 135 1.34 0.137126308
## 136 1.35 0.134411025
## 137 1.36 0.131749509
## 138 1.37 0.129140694
## 139 1.38 0.126583537
## 140 1.39 0.124077015
## 141 1.40 0.121620125
## 142 1.41 0.119211885
## 143 1.42 0.116851332
## 144 1.43 0.114537521
## 145 1.44 0.112269526
## 146 1.45 0.110046440
## 147 1.46 0.107867375
## 148 1.47 0.105731457
## 149 1.48 0.103637834
## 150 1.49 0.101585668
## 151 1.50 0.099574137
## 152 1.51 0.097602437
## 153 1.52 0.095669779
## 154 1.53 0.093775390
## 155 1.54 0.091918513
## 156 1.55 0.090098405
## 157 1.56 0.088314337
## 158 1.57 0.086565596
## 159 1.58 0.084851482
## 160 1.59 0.083171310
```

```
## 161 1.60 0.081524408
## 162 1.61 0.079910117
## 163 1.62 0.078327790
## 164 1.63 0.076776796
## 165 1.64 0.075256514
## 166 1.65 0.073766335
## 167 1.66 0.072305664
## 168 1.67 0.070873915
## 169 1.68 0.069470518
## 170 1.69 0.068094909
## 171 1.70 0.066746540
## 172 1.71 0.065424870
## 173 1.72 0.064129371
## 174 1.73 0.062859524
## 175 1.74 0.061614822
## 176 1.75 0.060394767
## 177 1.76 0.059198870
## 178 1.77 0.058026654
## 179 1.78 0.056877649
## 180 1.79 0.055751397
## 181 1.80 0.054647445
## 182 1.81 0.053565353
## 183 1.82 0.052504688
## 184 1.83 0.051465025
## 185 1.84 0.050445950
## 186 1.85 0.049447053
## 187 1.86 0.048467936
## 188 1.87 0.047508206
## 189 1.88 0.046567481
## 190 1.89 0.045645383
## 191 1.90 0.044741544
## 192 1.91 0.043855602
## 193 1.92 0.042987203
## 194 1.93 0.042135999
## 195 1.94 0.041301650
## 196 1.95 0.040483823
## 197 1.96 0.039682189
## 198 1.97 0.038896429
## 199 1.98 0.038126229
## 200 1.99 0.037371279
## 201 2.00 0.036631278
## 202 2.01 0.035905930
## 203 2.02 0.035194945
## 204 2.03 0.034498038
## 205 2.04 0.033814931
## 206 2.05 0.033145351
## 207 2.06 0.032489029
## 208 2.07 0.031845703
## 209 2.08 0.031215116
## 210 2.09 0.030597015
```

```
## 211 2.10 0.029991154
## 212 2.11 0.029397289
## 213 2.12 0.028815184
## 214 2.13 0.028244605
## 215 2.14 0.027685324
## 216 2.15 0.027137118
## 217 2.16 0.026599767
## 218 2.17 0.026073056
## 219 2.18 0.025556775
## 220 2.19 0.025050717
## 221 2.20 0.024554680
## 222 2.21 0.024068465
## 223 2.22 0.023591877
## 224 2.23 0.023124727
## 225 2.24 0.022666826
## 226 2.25 0.022217993
## 227 2.26 0.021778047
## 228 2.27 0.021346813
## 229 2.28 0.020924118
## 230 2.29 0.020509793
## 231 2.30 0.020103671
## 232 2.31 0.019705592
## 233 2.32 0.019315395
## 234 2.33 0.018932925
## 235 2.34 0.018558028
## 236 2.35 0.018190554
## 237 2.36 0.017830357
## 238 2.37 0.017477292
## 239 2.38 0.017131219
## 240 2.39 0.016791998
## 241 2.40 0.016459494
## 242 2.41 0.016133574
## 243 2.42 0.015814108
## 244 2.43 0.015500968
## 245 2.44 0.015194028
## 246 2.45 0.014893166
## 247 2.46 0.014598262
## 248 2.47 0.014309197
## 249 2.48 0.014025856
## 250 2.49 0.013748125
## 251 2.50 0.013475894
## 252 2.51 0.013209053
## 253 2.52 0.012947497
## 254 2.53 0.012691119
## 255 2.54 0.012439818
## 256 2.55 0.012193493
## 257 2.56 0.011952046
## 258 2.57 0.011715379
## 259 2.58 0.011483399
## 260 2.59 0.011256013
```

```
## 261 2.60 0.011033129
## 262 2.61 0.010814658
## 263 2.62 0.010600514
## 264 2.63 0.010390609
## 265 2.64 0.010184862
## 266 2.65 0.009983188
## 267 2.66 0.009785507
## 268 2.67 0.009591741
## 269 2.68 0.009401812
## 270 2.69 0.009215644
## 271 2.70 0.009033162
## 272 2.71 0.008854293
## 273 2.72 0.008678967
## 274 2.73 0.008507111
## 275 2.74 0.008338659
## 276 2.75 0.008173543
## 277 2.76 0.008011696
## 278 2.77 0.007853054
## 279 2.78 0.007697553
## 280 2.79 0.007545131
## 281 2.80 0.007395727
## 282 2.81 0.007249282
## 283 2.82 0.007105737
## 284 2.83 0.006965034
## 285 2.84 0.006827117
## 286 2.85 0.006691931
## 287 2.86 0.006559422
## 288 2.87 0.006429537
## 289 2.88 0.006302223
## 290 2.89 0.006177431
## 291 2.90 0.006055109
## 292 2.91 0.005935210
## 293 2.92 0.005817685
## 294 2.93 0.005702487
## 295 2.94 0.005589571
## 296 2.95 0.005478890
## 297 2.96 0.005370400
## 298 2.97 0.005264059
## 299 2.98 0.005159824
## 300 2.99 0.005057653
## 301 3.00 0.004957504
```

A.2. Table distribusi poisson untuk $x = 0,1,2,\ldots,20$

```
#In this case, we let X = 0, 1, ... , 20
#We still being able to compute X > 20 by using define_poisson function
arrival <- seq(0, 20, 1)
define_poisson <- function(X, lambda) {return(exp(-lambda) * (lambda ** X) /
factorial(X))}
define_cmltv_poisson <- function(r, lambda) {return(sum(define_poisson(seq(0,
```

```r
r, 1), lambda)))}
cmltv_prob_arrival <- c(sum(define_poisson(0:0, lambda = 8)))
for (arrivals in 2:length(arrival)) {
    cmltv_prob_arrival[arrivals] = sum(define_poisson(0:(arrivals-1), lambda
= 8))
}
df_poisson <- data.frame("arrival" = arrival[0:21], "poisson_probability" =
define_poisson(arrival, lambda = 8), "poisson_probability_sums" =
cmltv_prob_arrival[0:21])
print(df_poisson)
```

```
##    arrival poisson_probability poisson_probability_sums
## 1        0        0.0003354626             0.0003354626
## 2        1        0.0026837010             0.0030191637
## 3        2        0.0107348041             0.0137539677
## 4        3        0.0286261442             0.0423801120
## 5        4        0.0572522885             0.0996324005
## 6        5        0.0916036616             0.1912360621
## 7        6        0.1221382155             0.3133742775
## 8        7        0.1395865320             0.4529608095
## 9        8        0.1395865320             0.5925473414
## 10       9        0.1240769173             0.7166242587
## 11      10        0.0992615338             0.8158857926
## 12      11        0.0721902064             0.8880759990
## 13      12        0.0481268043             0.9362028033
## 14      13        0.0296164949             0.9658192982
## 15      14        0.0169237114             0.9827430096
## 16      15        0.0090259794             0.9917689890
## 17      16        0.0045129897             0.9962819787
## 18      17        0.0021237599             0.9984057386
## 19      18        0.0009438933             0.9993496319
## 20      19        0.0003974287             0.9997470606
## 21      20        0.0001589715             0.9999060321
```