

**PROYEK PRAKTIKUM STATISTIKA
DISTRIBUSI DATA KONTINYU**

DOSEN PENGAMPU : Ronny Susetyoko S. Si., M. Si



Disusun Oleh :

Reyna Aisyana
3321600002

PROGRAM STUDI SAINS DATA TERAPAN

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

DESEMBER 2021

LAPORAN PROYEK

❖ TUJUAN

Aplikasi berbasis R – Shiny yang dibuat memiliki tujuan untuk memperlihatkan hasil pengolahan dan visualisasi data dengan berbagai macam distribusi kontinu.

❖ METODOLOGI

a) Teoritis Sistematis

Distribusi peluang kontinu adalah peubah acak yang dapat memperoleh semua nilai pada skala kontinu. Ruang sampel kontinu adalah bila ruang sampel mengandung titik sampel yang tak terhingga banyaknya. Syarat dari distribusi kontinu adalah apabila fungsi $f(x)$ adalah fungsi padat peluang peubah acak kontinu X yang didefinisikan di atas himpunan semua bilangan riil R bila:

$$1. F(x) \geq 0 \text{ untuk semua } x \in R$$

$$2. \int_{-\infty}^{\infty} f(x) dx = 1$$

$$3. P(a < X < b) = \int_a^b f(x) dx$$

Distribusi peluang kontinu dibagi menjadi beberapa di antaranya :

1. DISTRIBUSI UNIFORM

Jika variabel random X memiliki nilai (kontinu) dengan kemungkinan kemunculan yang sama maka dikatakan bahwa variabel random (kontinu) x mengikuti distribusi uniform dengan fungsi densitas probabilitas.

2. DISTRIBUSI NORMAL

Distribusi normal disebut juga dengan Distribusi Gauss. Peubah acak (variabel random) pada distribusi normal merupakan peubah acak yang kontinu. Distribusi normal merupakan distribusi peluang yang paling sering digunakan dalam analisis statistik. Fungsi kepadatan peluang distribusi normal untuk peubah acak X adalah sebagai berikut:

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right)$$

Dimana x adalah peubah acak kontinu dan $-\infty \leq x \leq \infty$. Distribusi normal memiliki dua parameter yaitu mean μ dan varian σ^2 dimana $-\infty \leq \mu \leq \infty$ dan $\sigma^2 > 0$. Dengan demikian fungsi $f(x; \mu, \sigma^2)$ dapat dibaca bahwa peubah acak x mengikuti distribusi normal dengan rata-rata μ dan varian σ^2 , dan dapat ditulis menjadi $X \sim N(\mu, \sigma^2)$.

Berdasarkan rumus kepadatan peluang tersebut, bentuk dari kurva distribusi normal adalah lonceng (genta).

3. DISTRIBUSI EKSPONENSIAL

Distribusi Eksponensial termasuk distribusi kontinu dan merupakan bentuk khusus dari distribusi gamma dengan $\alpha = 1$. Distribusi eksponensial dapat ditandai dengan satu parameter, yakni θ . Setiap nilai unik θ menentukan distribusi eksponensial yang berbeda yang mana menghasilkan keluarga distribusi eksponensial. Rata-rata (mean) dari distribusi eksponensial adalah $\mu = \theta$ dan variannya yaitu $\sigma^2 = \theta^2$.

4. DISTRIBUSI WEIBULL

Distribusi Weibull adalah distribusi probabilitas yang kontinu. Distribusi Weibull banyak dipakai untuk memodelkan dan menganalisa data waktu kegagalan. Distribusi

Weibull memiliki parameter skala (scale) dan bentuk (shape), dimana kedua parameter tersebut lebih besar dari 0. Parameter-parameter tersebut harus diketahui agar dapat dicari lebih lanjut sifat dan karakteristik data yang berdistribusi Weibull. Fungsi kepadatan peluang distribusi Weibull :

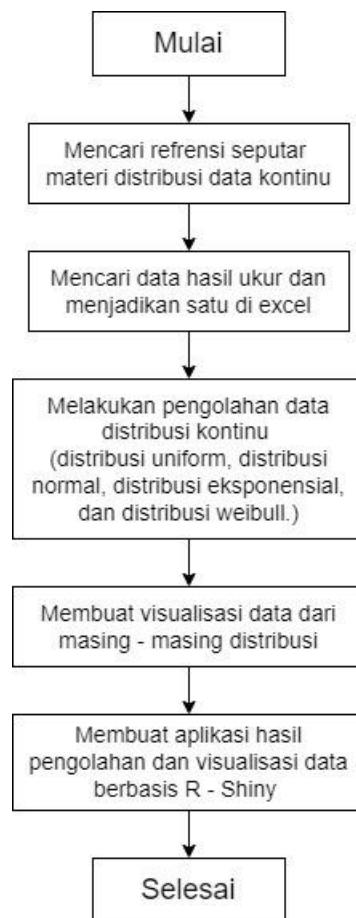
$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Mean dan variansi distribusi Weibull adalah :

$$\mu = \lambda \Gamma\left(1 + \frac{1}{k}\right)$$

$$\sigma^2 = \lambda^2 \left[\Gamma\left(1 + \frac{2}{k}\right) - \Gamma^2\left(1 + \frac{1}{k}\right) \right]$$

b) Langkah Kerja



❖ SUMBER DATA

Data : Data Temperature Maksimal Provinsi Jawa Timur Tahun 2020
 Sumber : Badan Meteorologi, Klimatologi, dan Geofisika
 Source : https://dataonline.bmkg.go.id/akses_data

❖ PENETAPAN VARIABEL

- a. TANGGAL : Tanggal pengukuran temperature maksimal (per hari)
- b. Temperature Maksimum : Temperature maksimal

❖ PREPROCESSING

- Import File

```
# import data
library(readxl)
dataTemp <- read_excel('C:/Users/HP CAMO/Documents/Statistika/TUGAS/DISTRIBUSI KONTINY
U/Data Temperature Jawa Timur.xlsx')
head(dataTemp)
```

Tanggal	Temperature Maksimum
<chr>	<dbl>
01-01-2020	28.2
02-01-2020	28.8
03-01-2020	28.8
04-01-2020	28.6
05-01-2020	28.3
06-01-2020	28.4
6 rows	

- Menghilangkan Nilai NA

```
temp <- na.omit(dataTemp$`Temperature Maksimum`)
head(temp)
[1] 28.2 28.8 28.8 28.6 28.3 28.4
```

❖ PEMODELAN

- Library

```
library(distrEx)
library(RcmdrMisc)
library(RColorBrewer)
library(ggplot2)
```

- Mean dan Variansi Fungsi Distribusi Kontinyu $f(x)$

```
# 1
f <- function(X)3*X^2
integrate(f, lower = 24.0, upper = 32.0)
18944 with absolute error < 2.1e-10
```

```
# 2
g <- AbscontDistribution(d = f, low1 = 0, up1 = 1)
p(g)(24.0) - p(g)(32)
[1] 0
```

```
# 3
E(g)
[1] 0.7496337
```

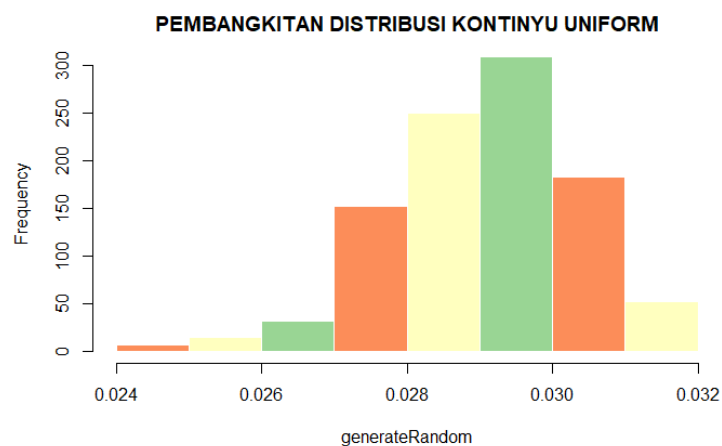
```
#4
var(g)
[1] 0.03768305
```

- Distribusi Uniform

```
# DISTRIBUSI UNIFORM
# Membangkitkan Bilangan Random Uniform Kontinyu
generateRandom <- sample(temp, size = 1000, replace = TRUE)
generateRandom <- generateRandom/1000
head(generateRandom)
[1] 0.0298 0.0278 0.0300 0.0285 0.0303 0.0312
```

```
tail(generateRandom)
[1] 0.0290 0.0294 0.0298 0.0266 0.0296 0.0308
```

```
# Histogram Pembangkitan Distribusi Uniform Dari Data Temperature Jatim 2020
histUniform<- hist(generateRandom,
  main = "PEMBANGKITAN DISTRIBUSI KONTINYU UNIFORM",
  #xlab = "Variable Random Dari Data Temperature Jawa Timur",
  breaks = c(0.024, 0.025, 0.026, 0.027, 0.028, 0.029, 0.030, 0.031, 0.032),
  xlim = c(0.024, 0.032),
  col = brewer.pal(3, "Spectral"),
  border = "white")
```



- Distribusi Normal – DNORM

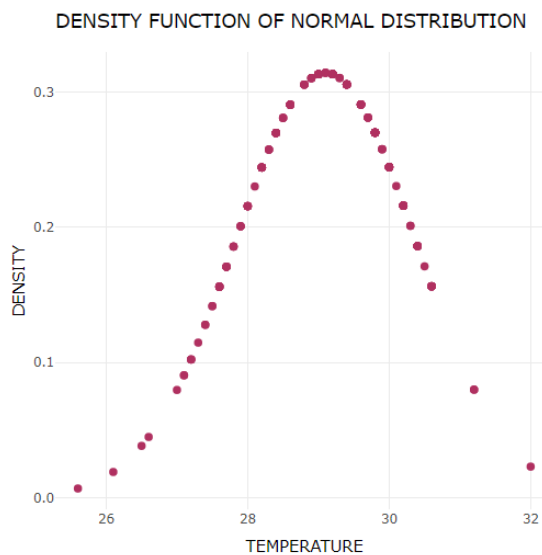
```
# DISTRIBUSI NORMAL - DNORM

# Fungsi dnorm() digunakan untuk memberikan ketinggian distribusi probabilitas pada setiap titik untuk mean dan deviasi standar tertentu.

# Membangkitkan Data Kontinyu Normal
sample.range <- temp[50:250]
temp.mean <- 29.10113
temp.sd <- 1.269193

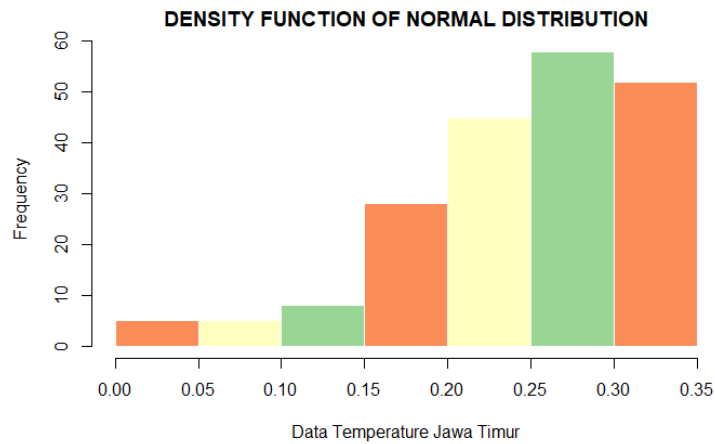
temp.dNorm <- dnorm(sample.range, mean = temp.mean, sd = temp.sd)
temp.df <- data.frame("Temperature" = sample.range, "Density" = temp.dNorm)

# Visualisasi Data Kontinyu Normal
plotdNorm <- ggplot(temp.df, aes(x = Temperature, y = Density)) +
  geom_point(col = "maroon") +
  labs(title = "DENSITY FUNCTION OF NORMAL DISTRIBUTION", x = "TEMPERATURE", y = "DENSITY") +
  theme_minimal()
ggplotly(plotdNorm)
```



```
histdNorm <- hist(temp.dNorm,
  main = "DENSITY FUNCTION OF NORMAL DISTRIBUTION",
  xlab = "Data Temperature Jawa Timur",
  #breaks = c(0.024, 0.025, 0.026, 0.027, 0.028, 0.029, 0.030, 0.031, 0.032),
  #xlim = c(0.024, 0.032),
```

```
col = brewer.pal(3, "Spectral"),
border = "white")
```



- Distribusi Normal – PORM

DISTRIBUSI NORMAL - PNORM

Fungsi pnorm() digunakan untuk memberikan probabilitas bilangan acak terdistribusi normal menjadi lebih kecil dari nilai bilangan tertentu. Ini juga disebut “Fungsi Distribusi Kumulatif”.

Membangkitkan Data Kontinyu Normal

```
sample.range <- temp[50:250]
```

```
temp.mean <- 29.10113
```

```
temp.sd <- 1.269193
```

```
cdf <- pnorm(sample.range, temp.mean, temp.sd, lower.tail = FALSE)
```

```
temp.df <- data.frame("Temperature" = sample.range, "Density" = temp.dNorm)
```

```
temp.df <- cbind(temp.df, "CDF_LowerTail" = cdf)
```

Visualisasi Data Kontinyu Normal

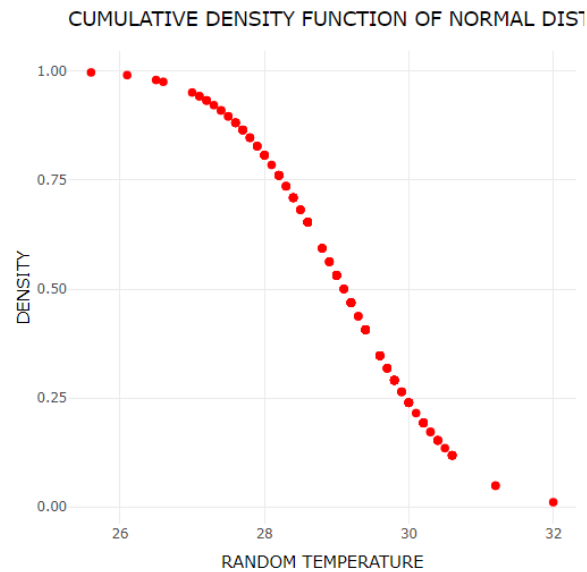
```
plotpNorm <- ggplot(temp.df, aes(x = Temperature, y = CDF_LowerTail)) +
```

```
  geom_point(col = "red") +
```

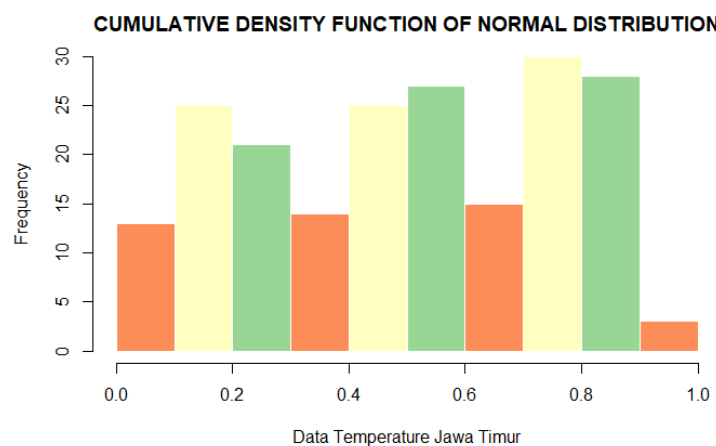
```
  labs(title = "CUMULATIVE DENSITY FUNCTION OF NORMAL DISTRIBUTION", x = "RANDOM TEMPERATURE", y = "DENSITY") +
```

```
  theme_minimal()
```

```
ggplotly(plotpNorm)
```



```
histpNorm<- hist(cdf,
  main = "CUMULATIVE DENSITY FUNCTION OF NORMAL DISTRIBUTION",
  xlab = "Data Temperature Jawa Timur",
  #breaks = c(0.024, 0.025, 0.026, 0.027, 0.028, 0.029, 0.030, 0.031, 0.032),
  #xlim = c(0.024, 0.032),
  col = brewer.pal(3, "Spectral"),
  border = "white")
```



- Distribusi Normal - QNORM

```
# Distribusi Probabilitas
x <- table(temp)/length(temp)
dPro <- data.frame(x)
head(dPro)
```


	temp <fctr>	Freq <dbl>
1	24.8	0.002832861
2	24.9	0.002832861
3	25	0.002832861
4	25.2	0.005665722
5	25.6	0.002832861
6	25.9	0.002832861

6 rows

DISTRIBUSI NORMAL - QNORM

Fungsi qnorm() digunakan untuk mengambil nilai probabilitas dan memberikan angka yang nilai kumulatifnya cocok dengan nilai probabilitas.

Membangkitkan Data Kontinu Normal

```
prob.range <- dPro$Freq
```

```
icdf.df <- data.frame("Probability" = prob.range, "Temperature" = qnorm(prob.range, temp.mean, temp.sd))
```

Visualisasi Data Kontinu Normal

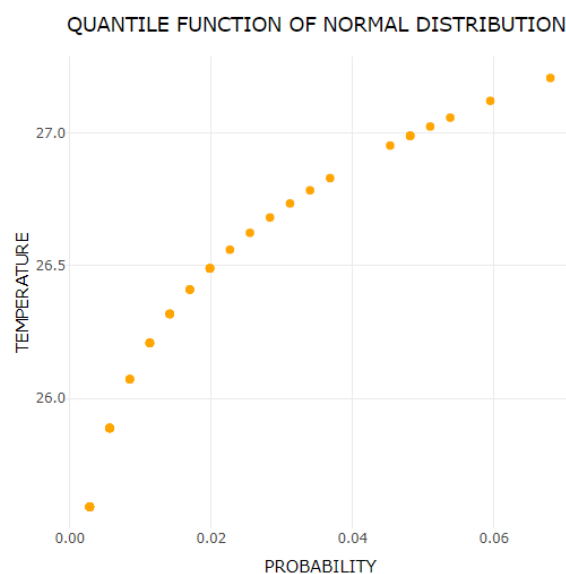
```
plotqNorm <- ggplot(icdf.df, aes(x = Probability, y = Temperature)) +
```

```
  geom_point(col = "orange") +
```

```
  labs(title = "QUANTILE FUNCTION OF NORMAL DISTRIBUTION", x = "PROBABILITY", y = "TEMPERATURE") +
```

```
  theme_minimal()
```

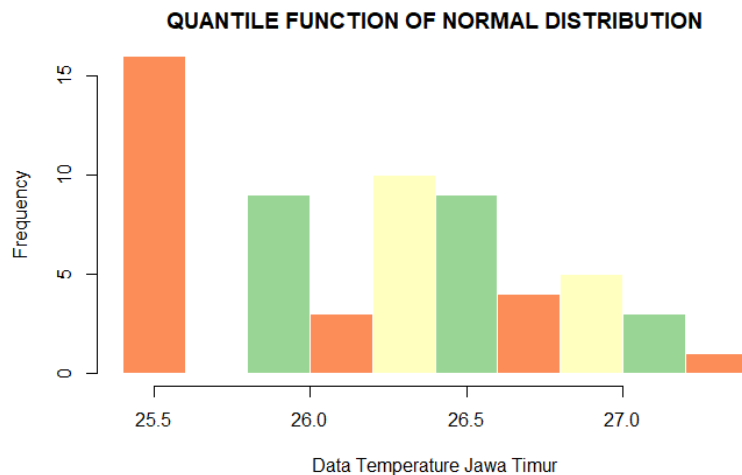
```
ggplotly(plotqNorm)
```



```

x <- table(temp)/length(temp)
dPro <- data.frame(x)
prob.range <- dPro$Freq
icdf.df <- qnorm(prob.range, temp.mean, temp.sd)
histqNormal<- hist(icdf.df,
  main = "QUANTILE FUNCTION OF NORMAL DISTRIBUTION",
  xlab = "Data Temperature Jawa Timur",
  col = brewer.pal(3, "Spectral"),
  border = "white")

```



- Distribusi Normal – RNORM

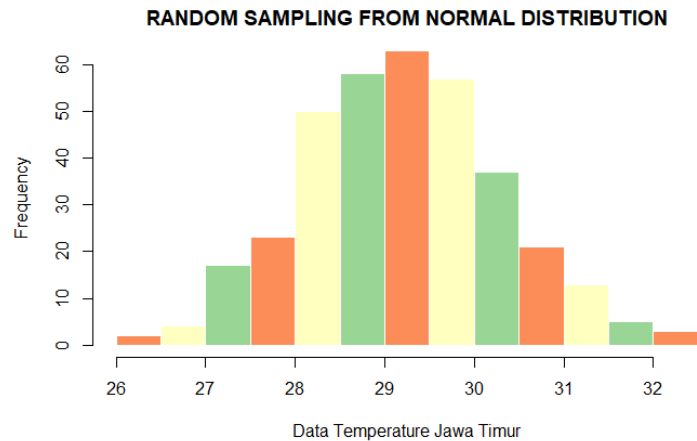
```

# DISTRIBUSI NORMAL - RNORM
# Fungsi rnorm() digunakan untuk menghasilkan bilangan acak yang distribusinya normal.

# Membangkitkan Data Kontinyu Normal
rNormal <- rnorm(353, temp.mean, temp.sd^0.5)

# Histogram Data Kontinyu Normal
histrNormal<- hist(rNormal,
  main = "RANDOM SAMPLING FROM NORMAL DISTRIBUTION",
  xlab = "Data Temperature Jawa Timur",
  col = brewer.pal(3, "Spectral"),
  border = "white")

```



- Distribusi Ekspensial – DEXP

```
# DISTRIBUSI EKSPONENSIAL - DEXP
# Membangkitkan Data Kontinyu Ekspensial

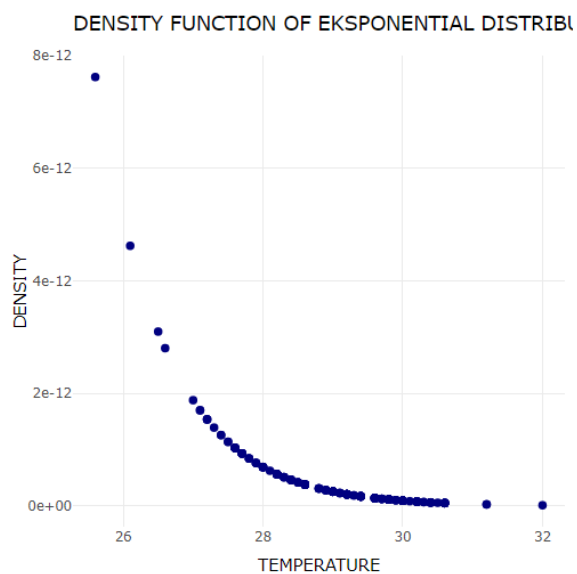
sample.range <- temp[50:250]

temp.dEks <- dexp(sample.range, rate = 1, log = FALSE)

temp.df <- data.frame("Temperature" = sample.range, "Density" = temp.dEks)

# Visualisasi Data Kontinyu Ekspensial

plotdEks <- ggplot(temp.df, aes(x = Temperature, y = Density)) +
  geom_point(col = "navy") +
  labs(title = "DENSITY FUNCTION OF EKSPONENTIAL DISTRIBUTION", x = "TEMPERATUR
E", y = "DENSITY")+
  theme_minimal()
ggplotly(plotdEks)
```



- Distribusi Eksponensial - PEXP

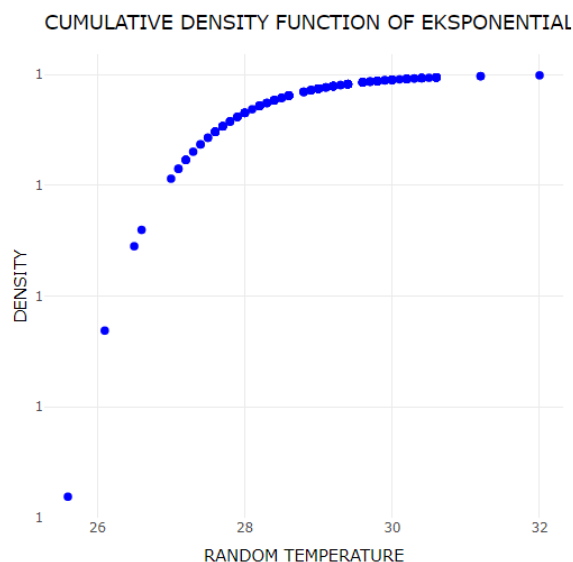
```
# DISTRIBUSI EKSPONENSIAL – PEXP
# Membangkitkan Data Kontinyu Eksponensial

sample.range <- temp[50:250]

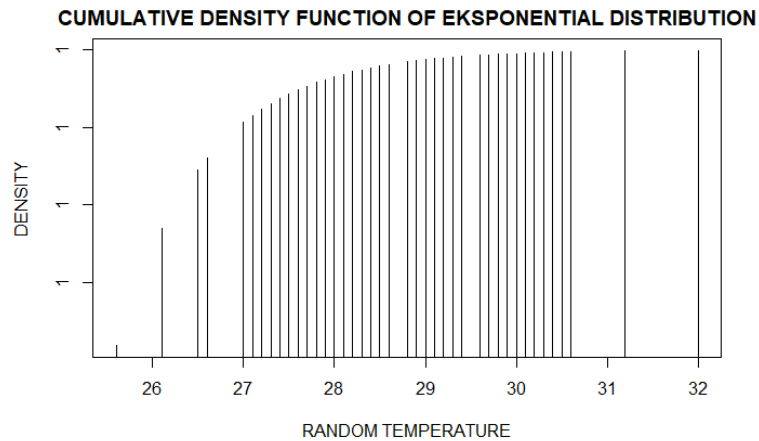
temp.pEks <- pexp(sample.range, lower.tail = TRUE, log.p = FALSE)
temp.df <- data.frame("Temperature" = sample.range, "Density" = temp.pEks)

# Visualisasi Data Kontinyu Eksponensial

plotpEks <- ggplot(temp.df, aes(x = Temperature, y = Density)) +
  geom_point(col = "blue") +
  labs(title = "CUMULATIVE DENSITY FUNCTION OF EKSPONENTIAL DISTRIBUTION", x = "
RANDOM TEMPERATURE", y = "DENSITY")+
  theme_minimal()
ggplotly(plotpEks)
```



```
plotdEksponensial <- plot(sample.range, temp.pEks, type = "h", xlab = "RANDOM TEMPERATURE"
, ylab = "DENSITY", main = "CUMULATIVE DENSITY FUNCTION OF EKSPONENTIAL DISTRI
BUTION")
```



- Distribusi Ekspensial – QEXP

```
# DISTRIBUSI EKSPONENSIAL - QEXP
# Membangkitkan Data Kontinyu Ekspensial

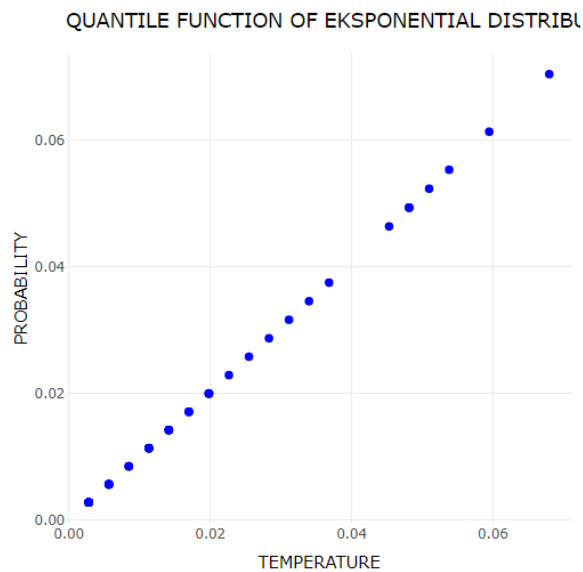
prob.range <- dPro$Freq

temp.qEks <- qexp(prob.range, lower.tail = TRUE, log.p = FALSE)

temp.df <- data.frame("Temperature" = prob.range, "Probability" = temp.qEks)

# Visualisasi Data Kontinyu Ekspensial

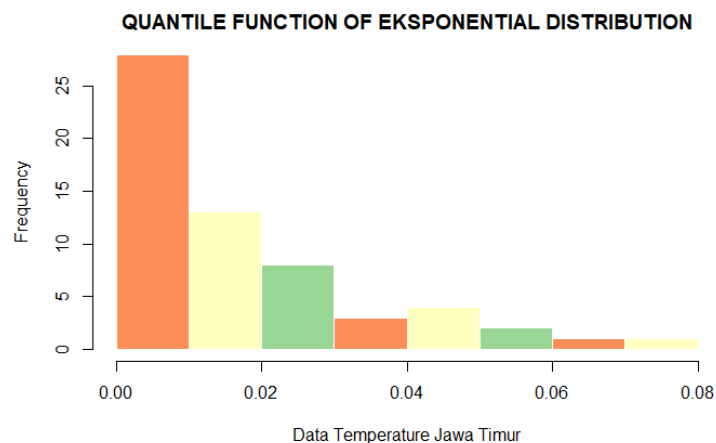
plotqEks <- ggplot(temp.df, aes(x = Temperature, y = Probability)) +
  geom_point(col = "blue") +
  labs(title = "QUANTILE FUNCTION OF EKSPONENTIAL DISTRIBUTION", x = "TEMPERATU
RE", y = "PROBABILITY")+
  theme_minimal()
ggplotly(plotqEks)
```



```

prob.range <- dPro$Freq
temp.qEks <- qexp(prob.range, lower.tail = TRUE, log.p = FALSE)
temp.df <- data.frame("Temperature" = prob.range, "Density" = temp.qEks)
histqEkspensial<- hist(temp.qEks,
  main = "QUANTILE FUNCTION OF EKSPONENTIAL DISTRIBUTION",
  xlab = "Data Temperature Jawa Timur",
  col = brewer.pal(3, "Spectral"),
  border = "white")

```



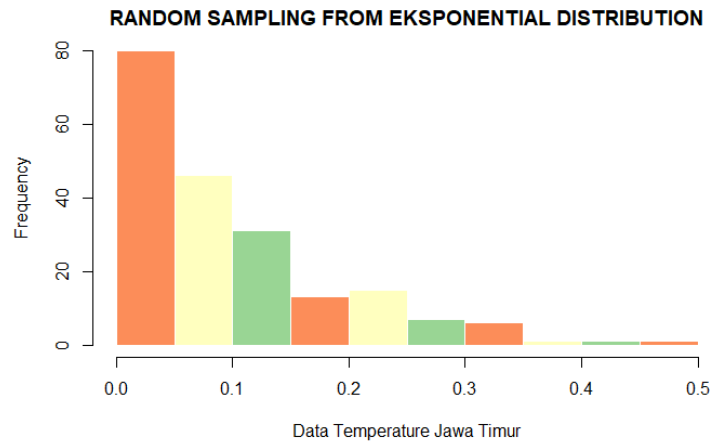
- Distribusi Ekspensial - REXP

```

# Membangkitkan Data Kontinyu Ekspensial
rEkspensial <- rexp(temp[50:250], 10)

# Histogram Data Kontinyu Normal
histEkspensial<- hist(rEkspensial,
  main = "RANDOM SAMPLING FROM EKSPONENTIAL DISTRIBUTION",
  xlab = "Data Temperature Jawa Timur",
  col = brewer.pal(3, "Spectral"),
  border = "white")

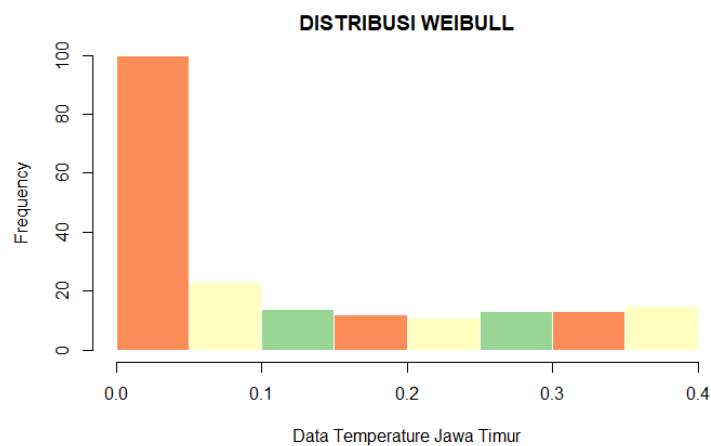
```



- Distribusi Weibull

```
# Membangkitkan Data Kontinyu Ekspensial
x <- temp[50:250]
y <- seq(25, 32)
weibull <- dweibull(x, shape = x, scale = y)

# Histogram Data Kontinyu Normal
histWeibull<- hist(weibull,
  main = "DISTRIBUSI WEIBULL",
  xlab = "Data Temperature Jawa Timur",
  col = brewer.pal(3, "Spectral"),
  border = "white")
```



R – Shiny Program

```
# R-SHINY PROGRAM
```

```
# DISTRIBUSI DATA KONTINYU
```

```
library(shiny)
library(shinydashboard)
library(rvest)
library(DT)
library(distrEx)
library(RcmdrMisc)
library(RColorBrewer)
library(ggplot2)
library(plotly)
library(glue)
```

```
# =====
```

```
ui <- dashboardPage(skin = "blue",
  dashboardHeader(title = "MAIN MENU"),
  dashboardSidebar(
    sidebarMenu(
      menuItem("HOME", icon = icon("house-user"),
        menuSubItem("Diagram",
          tabName = 'diagram',
          icon = icon('signal')),
        menuSubItem("Data",
          tabName = 'data',
          icon = icon('list-alt'))
      ),
      menuItem("DISTRIBUSI UNIFORM", tabName = "distribusiUniform", icon = icon("bezier-
curve")),
      menuItem("DISTRIBUSI NORMAL", icon = icon("bar-chart-o"),
        menuSubItem('Density Function',
          tabName = 'dNorm',
          icon = icon('ellipsis-v')),
        menuSubItem('Cumulative Density Function',
          tabName = 'pNorm',
          icon = icon('ellipsis-v')),
        menuSubItem('Quantile Function',
          tabName = 'qNorm',
          icon = icon('ellipsis-v')),
        menuSubItem('Random Sampling',
          tabName = 'rNorm',
          icon = icon('ellipsis-v'))
      ),
      menuItem("DISTRIBUSI EKSPONENSIAL", icon = icon("chart-pie"),
        menuSubItem('Density Function',
          tabName = 'dExp',
          icon = icon('ellipsis-v')),
        menuSubItem('Cumulative Density Function',
          tabName = 'pExp',
          icon = icon('ellipsis-v')),
        menuSubItem('Quantile Function',
          tabName = 'qExp',
          icon = icon('ellipsis-v'))
      )
    )
  )
)
```



```

        menuSubItem('Random Sampling',
            tabName = 'rExp',
            icon = icon('ellipsis-v'))
    ),
    menuItem("DISTRIBUSI WEIBULL", tabName = "distribusiWeibull", icon = icon("chart-area"))
)
),

# *****

dashboardBody(
  tabItems(
    tabItem("diagram",
      fluidPage(
        valueBox("TEMPERATURE", "VISUALISASI DATA TEMPERATURE MAKSIMUM PROVISI
JAWA TIMUR TAHUN 2020",
          icon = icon("temperature-high"), width = 12),
        tabBox(id = "menu1",
          tabPanel(title = "JANUARI", plotlyOutput(outputId = "plotJanuari"), width = 9),
          tabPanel(title = "FEBRUARI", plotlyOutput(outputId = "plotFebruari"), width = 9),
          tabPanel(title = "MARET", plotlyOutput(outputId = "plotMaret"), width = 9),
          tabPanel(title = "APRIL", plotlyOutput(outputId = "plotApril"), width = 9),
          tabPanel(title = "MEI", plotlyOutput(outputId = "plotMei"), width = 9),
          tabPanel(title = "JUNI", plotlyOutput(outputId = "plotJuni"), width = 9)
        ),
        tabBox(id = "menu2",
          tabPanel(title = "JULI", plotlyOutput(outputId = "plotJuli"), width = 9),
          tabPanel(title = "AGUSTUS", plotlyOutput(outputId = "plotAgustus"), width = 9),
          tabPanel(title = "SEPT", plotlyOutput(outputId = "plotSeptember"), width = 9),
          tabPanel(title = "OKT", plotlyOutput(outputId = "plotOktober"), width = 9),
          tabPanel(title = "NOV", plotlyOutput(outputId = "plotNovember"), width = 9),
          tabPanel(title = "DES", plotlyOutput(outputId = "plotDesember"), width = 9)
        )
      )),
    tabItem("data",
      fluidPage(
        box(title = "DATA TEMPERATURE MAKSIMUM PROVINSI JAWA TIMUR TAHUN 2020",
          status = "primary",
          solidHeader = T, width = 16, dataTableOutput("temperature"))
      )),
    # =====UNIFORM=====

    tabItem("distribusiUniform",
      fluidRow(
        box(title = "APA SIH DISTRIBUSI UNIFORM ITU?", status = "info", solidHeader = T,
          textOutput("penjelasanUniform", inline = F), width = 6),
        box(title = "PENJELASAN", status = "info", solidHeader = T,
          textOutput("noteUniform", inline = F), width = 6),
        box(title = "DATA TEMPERATURE PROVINSI JAWA TIMUR TAHUN 2020", status =
          "primary",
          background = "light-blue", plotOutput("histUniform"), width = 6),
        box(numericInput(inputId = "unif_input", "Tentukan Sample Data yang Ingin
          Diuji", value = 1000)),
        box(dataTableOutput("unif_hasil"))
      )),
  )
)

```

```
# =====NORMAL=====

        tabItem("dNorm",
            fluidRow(
                box(title = "APA SIH DISTRIBUSI NORMAL - DENSITY FUNCTION ITU?", status = "info",
solidHeader = T, textOutput("penjelasandNorm", inline = F), width = 6),
                box(title = "PENJELASAN", status = "info", solidHeader = T,
                    textOutput("notedNormal", inline = F), width = 6),
                tabBox(id = "tabdNormal",
                    tabPanel(title = "HISTOGRAM", plotOutput(outputId = "histdNormal"), width =
6),
                    tabPanel(title = "PLOT", plotlyOutput(outputId = "plotdNormal"), width = 6)),
                tabBox(id = "dNorm_input",
                    tabPanel(title = "BATAS BAWAH", numericInput(inputId = "dNorm_batasbawah",
                        "Tentukan Batas Bawah Data Sample [minimal 0]", value = 50)),
                    tabPanel(title = "BATAS ATAS", numericInput(inputId = "dNorm_batasatas",
                        "Tentukan Batas Atas Data Sample [maks 353]", value = 250)),
                    tabPanel(title = "RATA - RATA", numericInput(inputId =
"dNorm_mean", "Tentukan Nilai Mean yang Ingin Diuji", value = 29)),
                    tabPanel(title = "SIMPANGAN BAKU", numericInput(inputId =
"dNorm_sd", "Tentukan Nilai Simpangan Baku yang Ingin Diuji", value = 1.2))),
                box(dataTableOutput("dNorm_hasil"))
            )),
        tabItem("pNorm",
            fluidRow(
                box(title = "APA SIH DISTRIBUSI NORMAL - CUMULATIVE DENSITY FUNCTION ITU?",
status = "info", solidHeader = T, textOutput("penjelasnpNorm", inline = F), width = 12),
                box(title = "PENJELASAN", status = "info", solidHeader = T,
                    textOutput("notepNormal", inline = F), width = 12),
                tabBox(id = "tabpNormal",
                    tabPanel(title = "HISTOGRAM", plotOutput(outputId = "histpNormal"), width =
6),
                    tabPanel(title = "PLOT", plotlyOutput(outputId = "plotpNormal"), width = 6)),
                tabBox(id = "pNorm_input",
                    tabPanel(title = "BATAS BAWAH", numericInput(inputId = "pNorm_batasbawah",
                        "Tentukan Batas Bawah Data Sample [minimal 0]",
value = 50)),
                    tabPanel(title = "BATAS ATAS", numericInput(inputId = "pNorm_batasatas",
                        "Tentukan Batas Atas Data Sample [maks 353]", value =
250)),
                    tabPanel(title = "RATA - RATA", numericInput(inputId =
"pNorm_mean", "Tentukan Nilai Mean yang Ingin Diuji", value = 29)),
                    tabPanel(title = "SIMPANGAN BAKU", numericInput(inputId =
"pNorm_sd", "Tentukan Nilai Simpangan Baku yang Ingin Diuji", value = 1.2))),
                box(dataTableOutput("pNorm_hasil"))
            )),
        tabItem("qNorm",
            fluidRow(
                box(title = "APA SIH DISTRIBUSI NORMAL - QUANTILE FUNCTION ITU?", status =
"info", solidHeader = T, textOutput("penjelasnqNorm", inline = F), width = 6),
                box(title = "PENJELASAN", status = "info", solidHeader = T,
                    textOutput("noteqNormal", inline = F), width = 6),
                tabBox(id = "tabqNormal",
                    tabPanel(title = "HISTOGRAM", plotOutput(outputId = "histqNormal"), width =
6),
                    tabPanel(title = "PLOT", plotlyOutput(outputId = "plotqNormal"), width = 6)),
                tabBox(id = "qNorm_input",
```

```

        tabPanel(title = "RATA - RATA", numericInput(inputId =
"qNorm_mean", "Tentukan Nilai Mean yang Ingin Diuji", value = 29)),
        tabPanel(title = "SIMPANGAN BAKU", numericInput(inputId =
"qNorm_sd", "Tentukan Nilai Simpangan Baku yang Ingin Diuji", value = 1.2))),
        box(dataTableOutput("qNorm_hasil"))
    )),
    tabItem("rNorm",
        fluidRow(
            box(title = "APA SIH DISTRIBUSI NORMAL - RANDOM FUNCTION ITU?", status =
"info", solidHeader = T, textOutput("penjelasnrNorm", inline = F), width = 6),
            box(title = "PENJELASAN", status = "info", solidHeader = T,
                textOutput("noterNormal", inline = F), width = 6),
            box(title = "DATA TEMPERATURE PROVINSI JAWA TIMUR TAHUN 2020", status =
"primary",
                background = "light-blue", plotOutput("histrNormal"), width = 6),
            tabBox(id = "rNorm_input",
                tabPanel(title = "RATA - RATA", numericInput(inputId =
"rNorm_mean", "Tentukan Nilai Mean yang Ingin Diuji", value = 29)),
                tabPanel(title = "SIMPANGAN BAKU", numericInput(inputId =
"rNorm_sd", "Tentukan Nilai Simpangan Baku yang Ingin Diuji", value = 1.2))),
                box(dataTableOutput("rNorm_hasil"))
            )),

# =====EKSPONENSIAL=====

        tabItem("dExp",
            fluidRow(
                box(title = "APA SIH DISTRIBUSI EKSPONENSIAL - DENSITY FUNCTION ITU?", status =
"info", solidHeader = T, textOutput("penjelasdExp", inline = F), width = 6),
                box(title = "PENJELASAN", status = "info", solidHeader = T,
                    textOutput("notedExp", inline = F), width = 6),
                box(title = "DATA TEMPERATURE PROVINSI JAWA TIMUR TAHUN 2020", status =
"primary", solidHeader = T,
                    background = "light-blue", plotlyOutput(outputId = "plotdExp"), width = 6),
                tabBox(id = "dExp_input",
                    tabPanel(title = "BATAS BAWAH", numericInput(inputId = "dExp_batasbawah",
                        "Tentukan Batas Bawah Data Sample [minimal 0]",
                        value = 50)),
                    tabPanel(title = "BATAS ATAS", numericInput(inputId = "dExp_batasatas",
                        "Tentukan Batas Atas Data Sample [maks 353]", value =
250)),
                    tabPanel(title = "RATE", numericInput(inputId = "dExp_rate", "Tentukan Nilai
RATE yang Ingin Diuji", value = 1))),
                box(dataTableOutput("dExp_hasil"))
            )),
        tabItem("pExp",
            fluidRow(
                box(title = "APA SIH DISTRIBUSI EKSPONENSIAL - CUMULATIVE DENSITY FUNCTION
ITU?", status = "info", solidHeader = T, textOutput("penjelaspExp", inline = F), width = 6),
                box(title = "PENJELASAN", status = "info", solidHeader = T,
                    textOutput("notepExp", inline = F), width = 6),
                tabBox(id = "tabpExp",
                    tabPanel(title = "HISTOGRAM", plotOutput(outputId = "histpExp"), width = 9),
                    tabPanel(title = "PLOT", plotlyOutput(outputId = "plotpExp"), width = 9)),
                tabBox(id = "pExp_input",
                    tabPanel(title = "BATAS BAWAH", numericInput(inputId = "pExp_batasbawah",

```

```

                                "Tentukan Batas Bawah Data Sample [minimal 0]", value
= 50)),
                                tabPanel(title = "BATAS ATAS", numericInput(inputId = "pExp_batasatas",
                                "Tentukan Batas Atas Data Sample [maks 353]", value =
250))),
                                box(dataTableOutput("pExp_hasil"))
                                )),
                                tabItem("qExp",
                                fluidRow(
                                box(title = "APA SIH DISTRIBUSI EKSPONENSIAL - QUANTILE FUNCTION ITU?", status
= "info", solidHeader = T, textOutput("penjelasanqExp", inline = F), width = 6),
                                box(title = "PENJELASAN", status = "info", solidHeader = T,
                                textOutput("noteqExp", inline = F), width = 6),
                                box(title = "DATA TEMPERATURE PROVINSI JAWA TIMUR TAHUN 2020",status =
"primary", solidHeader = T,
                                background = "light-blue", plotOutput(outputId = "histqExp"), width = 6),
                                box(sliderInput("qExp_slider", label = "Slider Input", min = 0, max = 20, value = 6,
step = 0,5), width = 6),
                                box(dataTableOutput("qExp_hasil"))
                                )),
                                tabItem("rExp",
                                fluidRow(
                                box(title = "APA SIH DISTRIBUSI EKSPONENSIAL - RANDOM FUNCTION ITU?", status =
"info", solidHeader = T, textOutput("penjelasanrExp", inline = F), width = 6),
                                box(title = "PENJELASAN", status = "info", solidHeader = T,
                                textOutput("noterExp", inline = F), width = 6),
                                box(title = "DATA TEMPERATURE PROVINSI JAWA TIMUR TAHUN 2020",status =
"primary", solidHeader = T,
                                background = "light-blue", plotOutput(outputId = "histrExp"), width = 6),
                                tabBox(id = "rExp_input",
                                tabPanel(title = "BATAS BAWAH", numericInput(inputId = "rExp_batasbawah",
                                "Tentukan Batas Bawah Data Sample [minimal 0]",
value = 50)),
                                tabPanel(title = "BATAS ATAS", numericInput(inputId = "rExp_batasatas",
                                "Tentukan Batas Atas Data Sample [maks 353]", value =
250)),
                                tabPanel(title = "LAMBDA", numericInput(inputId = "rExp_lambda",
                                "Tentukan Lambda", value = 10))),
                                box(dataTableOutput("rExp_hasil"))
                                )),

```

=====WEIBULL=====

```

                                tabItem("distribusiWeibull",
                                box(title = "APA SIH DISTRIBUSI WEIBULL ITU?", status = "info", solidHeader = T,
textOutput("penjelasanWeibull", inline = F), width = 6),
                                box(title = "PENJELASAN", status = "info", solidHeader = T,
                                textOutput("noteWeibull")),
                                box(title = "DATA TEMPERATURE PROVINSI JAWA TIMUR TAHUN 2020", status =
"primary", solidHeader = T,
                                background = "light-blue", plotOutput(outputId = "histWeibull")),
                                tabBox(id = "weibull_input",
                                tabPanel(title = "BATAS BAWAH", numericInput(inputId = "weibull_batasbawah",

```

```

                    "Tentukan Batas Bawah Data Sample [minimal 0]", value =
50)),
                tabPanel(title = "BATAS ATAS", numericInput(inputId = "weibull_batasatas",
                    "Tentukan Batas Atas Data Sample [maks 353]", value =
250)),
                tabPanel(title = "MINIMAL", numericInput(inputId = "weibull_min", "Tentukan Nilai
Sequence Minimal", value = 25)),
                tabPanel(title = "MAKSIMAL", numericInput(inputId = "weibull_maks", "Tentukan
Nilai Sequence Maksimal", value = 32))),
                box(dataTableOutput("weibull_hasil")))
            ) # kurung tutup tabItems
        ) # kurung tutup dashboardBody
    ) # kurung tutup dashboardPage

```

```
# R-SHINY PROGRAM (SERVER.UI)
```

```
server <- function(input, output){
```

```
  output$noteWeibull <- renderText({
```

```
    "Parameter scale yang digunakan dalam pengujian distribusi weibull adalah Data Temperature  
Maksimal Provinsi Jawa Timur Tahun 2020
```

```
    dimana batas bawah dan batas atas data yang digunakan sebagai sample ditentukan oleh user.
```

```
    Sedangkan parameter shape merupakan sequence yang nilai minimal dan maksimalnya juga ditentukan  
oleh use."
```

```
  )
```

```
  output$penjelasanWeibull <- renderText({
```

```
    "Distribusi Weibull adalah distribusi probabilitas kontinu yang banyak digunakan untuk memodelkan  
dan menganalisa data waktu kegagalan.
```

```
    Pada perhitungan distribusi weibull memiliki parameter skala dan shape."
```

```
  )
```

```
  output$histWeibull <- renderPlot({
```

```
    x <- temp[input$weibull_batasbawah:input$weibull_batasatas]
```

```
    y <- seq(input$weibull_min, input$weibull_maks)
```

```
    weibull <- dweibull(x, shape = x, scale = y)
```

```
    histWeibull <- hist(weibull,  
      main = "DISTRIBUSI WEIBULL",  
      xlab = "Data Temperature Jawa Timur",  
      col = brewer.pal(5, "Pastel2"),  
      border = "white")
```

```
  })
```

```
  output$weibull_hasil <- renderDataTable({
```

```
    x <- temp[input$weibull_batasbawah:input$weibull_batasatas]
```

```
    y <- seq(input$weibull_min, input$weibull_maks)
```

```
    weibull <- dweibull(x, shape = x, scale = y)
```

```
    weibull_tabel <- data.frame(seq(1, length(x)), weibull)
```

```
    colnames(weibull_tabel) <- c("SEQUENCE", "PROBABILITAS HASIL WEIBULL")
```

```
    expr = weibull_tabel
```

```
  })
```

```
# =====DISTRIBUSI EKSPONENSIAL [R - EXP]=====
```

```
  output$noterExp <- renderText( # BELUM NEMU YG BAGUS
```

```
    "Dalam pengujian random sampling distribusi eksponensial, sample data diambil dari Data Temperature  
Maksimal Provinsi Jawa Timur Tahun 2020
```

```
    dimana batas bawah dan batas atas untuk sample data ditentukan oleh user. Parameter lain yang  
digunakan adalah rate atau lambda yang diset
```

```
    dengan nilai 10, namun nilai ini dapat diubah sesuai keinginan user."
```

```
  )
```

```
  output$penjelasanrExp <- renderText( # KURANG KATA - KATANYA
```

```
    "Fungsi random sampling distribusi eksponensial merupakan fungsi yang digunakan untuk  
mensimulasikan sekumpulan bilangan acak yang diambil  
dari distribusi eksponensial."
```

```
  )
```

```
  output$histrExp <- renderPlot({
```

```
    rExp_sample <- temp[input$rExp_batasbawah:input$rExp_batasatas]
```

```
    rEkspensial <- rexp(rExp_sample, input$rExp_lambda)
```

```
    histEkspensial <- hist(rEkspensial,  
      main = "RANDOM SAMPLING FROM EXPONENTIAL DISTRIBUTION",  
      xlab = "Data Temperature Jawa Timur",  
      col = brewer.pal(9, "Pastel1"),  
      border = "white")
```

```
  })
```

```

output$rExp_hasil <- renderDataTable({
  rExp_sample <- temp[input$rExp_batasbawah:input$rExp_batasatas]
  rEksponensial <- rexp(rExp_sample, input$rExp_lambda)
  rEks_tabel <- data.frame(rExp_sample, rEksponensial)
  colnames(rEks_tabel) <- c("TEMPERATURE SAMPLE", "PROBABILITAS HASIL R - EXP")
  expr = rEks_tabel

})

# =====DISTRIBUSI EKSPONENSIAL [Q - EXP]=====
output$noteqExp <- renderText(
  "Nilai value berupa quantile yang digunakan berasal dari hasil perhitungan peluang data.
  Paramater lower.tail yang bernilai TRUE digunakan agar pengujian yang dilakukan akan
  mengembalikan nilai dari sisi sebelah kiri nilai value dan parameter
  log.p yang diset dengan nilai FALSE."
)
output$penjelasanqExp <- renderText(
  "Quantile function dari distribusi eksponensial merupakan fungsi yang akan menghasilkan nilai
  kemungkinan atau peluang yang sesuai dengan
  nilai quantile yang telah ditentukan."
)
output$histqExp <- renderPlot({
  prob.range <- dPro$Freq
  temp.qEks <- qexp(prob.range, lower.tail = TRUE, log.p = FALSE)
  temp.df <- data.frame("Temperature" = prob.range, "Density" = temp.qEks)
  histqEksponensial<- hist(temp.qEks,
    main = "QUANTILE FUNCTION OF EXPONENTIAL DISTRIBUTION",
    xlab = "Data Temperature Jawa Timur",
    breaks = input$qExp_slider,
    col = brewer.pal(3, "Pastel1"),
    border = "white")
})
output$qExp_hasil <- renderDataTable({
  prob.range <- dPro$Freq
  temp.qEks <- qexp(prob.range, lower.tail = TRUE, log.p = FALSE)
  qEks_tabel <- data.frame(seq(1,60), temp.qEks)
  colnames(qEks_tabel) <- c("SEQUENCE", "PROBABILITAS HASIL Q - EXP")
  expr = qEks_tabel

})

# =====DISTRIBUSI EKSPONENSIAL [P - EXP]=====
output$notepExp <- renderText(
  "Dataset yang digunakan adalah Data Temperature Maksimal Provinsi Jawa Timur Tahun 2020 dengan
  sample data yang diuji sesuai dengan input user.
  Dengan tambahan parameter lower.tail yang bernilai TRUE, artinya nilai yang dikembalikan berada di
  sebelah kiri nilai value. Dan parameter
  log.p yang menyimpan nilai FALSE."
)
output$penjelasanpExp <- renderText(
  "Cumulative Density Function dari distribusi eksponensial merupakan fungsi yang akan mengembalikan
  nilai yang sesuai
  dari fungsi distribusi kumulatif eksponensial untuk vektor quantile yang telah ditentukan."
)
output$plotpExp <- renderPlotly({
  sample.range <- temp[input$pExp_batasbawah:input$pExp_batasatas]
  temp.pEks <- pexp(sample.range, lower.tail = TRUE, log.p = FALSE)

```

```

temp.df <- data.frame("Temperature" = sample.range, "Density" = temp.pEks)
plotpEks <- ggplot(temp.df, aes(x = Temperature, y = Density)) +
  geom_point(col = "skyblue") +
  labs(title = "CUMULATIVE DENSITY FUNCTION OF EXPONENTIAL DISTRIBUTION", x = "RANDOM
TEMPERATURE", y = "DENSITY")+
  theme_minimal()
ggplotly(plotpEks)
})
output$histpExp <- renderPlot({
  sample.range <- temp[input$pExp_batasbawah:input$pExp_batasatas]
  temp.pEks <- pexp(sample.range, lower.tail = TRUE, log.p = FALSE)
  plotdEkspensial <- plot(sample.range, temp.pEks, type = "h", xlab = "RANDOM TEMPERATURE", ylab
= "DENSITY",
                        main = "CUMULATIVE DENSITY FUNCTION OF EKSPONENTIAL DISTRIBUTION")
})
output$pExp_hasil <- renderDataTable({
  sample.range <- temp[input$pExp_batasbawah:input$pExp_batasatas]
  temp.pEks <- pexp(sample.range, lower.tail = TRUE, log.p = FALSE)
  pEks_tabel <- data.frame(sample.range, temp.pEks)
  colnames(pEks_tabel) <- c("TEMPERATURE SAMPLE", "PROBABILITAS HASIL P - EXP")
  expr = pEks_tabel

})

# =====DISTRIBUSI EKSPONENSIAL [D - EXP]=====
output$notedExp <- renderText(
  "Parameter yang dibutuhkan untuk uji densitas distribusi eksponensial adalah nilai value berupa
sample temperature dari Data Temperature Maksimal
Provinsi Jawa Timur Tahun 2020, nilai rate, dan log. User dapat menentukan sample data dan nilai rate
yang ingin diuji. Namun, nilai log akan
diset dengan nilai FALSE."
)
output$penjelasandExp <- renderText(
  "Fungsi density distribusi eksponensial merupakan fungsi yang akan mengembalikan nilai yang sesuai
dari kerapatan eksponensial
untuk vektor dari quantile yang telah ditentukan."
)
output$plotdExp <- renderPlotly({
  sample.range <- temp[input$dExp_batasbawah:input$dExp_batasatas]
  temp.dEks <- dexp(sample.range, rate = input$dExp_rate, log = FALSE)
  temp.df <- data.frame("Temperature" = sample.range, "Density" = temp.dEks)
  plotdEks <- ggplot(temp.df, aes(x = Temperature, y = Density)) +
    geom_point(col = "navy") +
    labs(title = "DENSITY FUNCTION OF EXPONENTIAL DISTRIBUTION", x = "TEMPERATURE", y =
"DENSITY")+
    theme_minimal()
  ggplotly(plotdEks)
})
output$dExp_hasil <- renderDataTable({
  sample.range <- temp[input$dExp_batasbawah:input$dExp_batasatas]
  temp.dEks <- dexp(sample.range, rate = input$dExp_rate, log = FALSE)
  dEks_tabel <- data.frame(sample.range, temp.dEks)
  colnames(dEks_tabel) <- c("TEMPERATURE SAMPLE", "PROBABILITAS HASIL D - EXP")
  expr = dEks_tabel

})

```



```
# =====DISTRIBUSI NORMAL [R - NORM]=====
output$noterNormal <- renderText(
  "Parameter uji yang digunakan adalah nilai value, mean, dan simpangan baku dari Data Temperature
Maksimal Provinsi Jawa Timur Tahun 2020.
  Hasil dari uji random sampling akan berupa vektor dari 353 variabel acak berdistribusi normal dengan
nilai rata - rata dan simpangan sesuai
  input dari user."
)
output$penjelasanrNorm <- renderText(
  "Fungsi random sampling digunakan untuk menggambarkan sample acak dari distribusi normal. Fungsi
ini akan menghasilkan vektor acak berdistribusi normal
  dengan panjang vektor n, mean populasi, dan simpangan baku populasi."
)
output$histrNormal <- renderPlot({
  rNorm_mean <- input$rNorm_mean
  rNorm_sd <- input$rNorm_sd
  rNormal <- rnorm(353, rNorm_mean, rNorm_sd^0.5)
  histrNormal<- hist(rNormal,
    main = "RANDOM SAMPLING FROM NORMAL DISTRIBUTION",
    xlab = "Data Temperature Jawa Timur",
    col = brewer.pal(9, "PuBuGn"),
    border = "white")
})
output$rNorm_hasil <- renderDataTable({
  rNorm_mean <- input$rNorm_mean
  rNorm_sd <- input$rNorm_sd
  rNormal <- rnorm(353, rNorm_mean, rNorm_sd^0.5)
  rNorm_tabel <- data.frame(seq(1,353), rNormal)
  colnames(rNorm_tabel) <- c("VEKTOR", "PROBABILITAS HASIL R - NORM")
  expr = rNorm_tabel
})

# =====DISTRIBUSI NORMAL [Q - NORM]=====
output$noteqNormal <- renderText(
  "Pada uji quantile distribusi normal, nilai value yang digunakan diambil dari hasil perhitungan peluang
berdasarkan
  Data Temperature Maksimal Provinsi Jawa Timur Tahun 2020. Dengan parameter lain berupa nilai rata
- rata dan simpangan
  baku yang dapat diatur oleh keinginan user."
)
output$penjelasanqNorm <- renderText(
  "Fungsi quantile pada distribusi normal merupakan kebalikan atau invers dari fungsi kepadatan
kumulatif (CDF). Sehingga,
  hasil dari uji quantile ini akan memetakan dari probabilitas ke nilai value."
)
output$plotqNormal <- renderPlotly({
  prob.range <- dPro$Freq
  qNorm_mean <- input$qNorm_mean
  qNorm_sd <- input$qNorm_sd
  icdf.df <- data.frame("Probability" = prob.range, "Temperature" = qnorm(prob.range, qNorm_mean,
qNorm_sd))
  plotqNorm <- ggplot(icdf.df, aes(x = Probability, y = Temperature)) +
    geom_point(col = "orange") +
    labs(title = "QUANTILE FUNCTION OF NORMAL DISTRIBUTION", x = "PROBABILITY", y =
"TEMPERATURE")+
    theme_minimal()
})
```

```

    ggplotly(plotqNorm)
  })
  output$histqNormal <- renderPlot({
    qNorm_mean <- input$qNorm_mean
    qNorm_sd <- input$qNorm_sd
    x <- table(temp)/length(temp)
    dPro <- data.frame(x)
    prob.range <- dPro$Freq
    icdf.df <- qnorm(prob.range, qNorm_mean, qNorm_sd)
    histqNormal<- hist(icdf.df,
      main = "QUANTILE FUNCTION OF NORMAL DISTRIBUTION",
      xlab = "Data Temperature Jawa Timur",
      col = brewer.pal(7, "PuBuGn"),
      border = "white")
  })
  output$qNorm_hasil <- renderDataTable({
    prob.range <- dPro$Freq
    qNorm_mean <- input$qNorm_mean
    qNorm_sd <- input$qNorm_sd
    hitung <- qnorm(prob.range, qNorm_mean, qNorm_sd)
    qNorm_tabel <- data.frame(prob.range, hitung)
    colnames(qNorm_tabel) <- c("TEMPERATURE SAMPLE", "PROBABILITAS HASIL Q - NORM")
    expr = qNorm_tabel
  })

# =====DISTRIBUSI NORMAL [P - NORM]=====
  output$notepNormal <- renderText(
    "Pada pengujian cumulative density function sample data yang digunakan diambil dari Data
    Temperature
    Maksimal Provinsi Jawa Timur Tahun 2020, dimana batas bawah dan batas atas data sample dapat
    ditentukan oleh user begitu juga
    dengan parameter lain seperti, nilai rata - rata dan simpangan baku. Namun pada setelan awal sample
    data akan diambil dari dataset
    urutan ke 50 hingga 250 dengan nilai rata - rata = 29 dan simpangan baku = 1,2 dan ditambah dengan
    parameter lower.tail yang diset
    sesuai default yaitu TRUE. Nantinya plot yang dihasilkan akan bergerak mendekati nilai nol."
  )
  output$penjelasanpNorm <- renderText(
    "Cumulative Density Function (CDF) adalah fungsi kepadatan kumulatif dari distribusi normal suatu
    variabel acak tertentu.
    Hasil uji kumulatif akan sama dengan yang diperoleh dari menjumlahkan secara manual probabilitas
    yang diperoleh
    melalui uji densitas distribusi normal. Terdapat penambahan argumen lower.tail yang digunakan
    sebagai parameter di uji kumulatif,
    lower.tail sendiri digunakan untuk mengatur area mana yang ingin dilakukan uji kumulatif. Secara
    default lower.tail akan bernilai
    TRUE yang artinya akan mengembalikan area sebelah kiri nilai value yang diberikan dalam distribusi
    normal."
  )
  output$plotpNormal <- renderPlotly({
    sample.range <- temp[input$pNorm_batasbawah:input$pNorm_batasatas]
    temp.mean <- input$pNorm_mean
    temp.sd <- input$pNorm_sd
    cdf <- pnorm(sample.range, temp.mean, temp.sd, lower.tail = TRUE)
    temp.df <- data.frame("Temperature" = sample.range, "Density" = temp.dNorm)
    temp.df <- cbind(temp.df, "CDF_LowerTail" = cdf)
  })

```

```

    plotpNorm <- ggplot(temp.df, aes(x = Temperature, y = CDF_LowerTail)) +
      geom_point(col = "red") +
      labs(title = "CUMULATIVE DENSITY FUNCTION OF NORMAL DISTRIBUTION", x = "RANDOM
TEMPERATURE", y = "DENSITY")+
      theme_minimal()
    ggplotly(plotpNorm)
  })
  output$histpNormal <- renderPlot({
    pNorm.sample <- temp[input$pNorm_batasbawah:input$pNorm_batasatas]
    temp.mean <- input$pNorm_mean
    temp.sd <- input$pNorm_sd
    cdf <- pnorm(pNorm.sample, temp.mean, temp.sd, lower.tail = TRUE)
    histpNorm<- hist(cdf,
      main = "CUMULATIVE DENSITY FUNCTION OF NORMAL DISTRIBUTION",
      xlab = "Data Temperature Jawa Timur",
      col = brewer.pal(5, "PuBuGn"),
      border = "white")
  })
  output$pNorm_hasil <- renderDataTable({
    sample.range <- temp[input$pNorm_batasbawah:input$pNorm_batasatas]
    temp.mean <- input$pNorm_mean
    temp.sd <- input$pNorm_sd
    cdf <- pnorm(sample.range, temp.mean, temp.sd, lower.tail = TRUE)
    pNorm_tabel <- data.frame(sample.range, cdf)
    colnames(pNorm_tabel) <- c("TEMPERATURE SAMPLE", "PROBABILITAS HASIL P - NORM")
    expr = pNorm_tabel

  })

# =====DISTRIBUSI NORMAL [D - NORM]=====
  output$notedNormal <- renderText(
    "Pada pengujian densitas distribusi normal, digunakan sample Data Temperature Maksimal Provinsi
Jawa Timur Tahun 2020.
    Dimana batas bawah dan batas atas data yang akan digunakan sebagai sample ditentukan berdasarkan
input user, begitu juga
    dengan nilai rata - rata dan simpangan baku yang diperlukan sebagai parameter pegujian."
  )
  output$penjelasandNorm <- renderText( # KURANG KATA - KATANYA
    "Probability Density Function atau distribusi densitas menunjukkan nilai probabilitas
dari sebuah data yang diamati dengan pengukuran nilai tertentu. Pada uji densitas normal
digunakan parameter x sebagai value, miu yang mendefinisikan mean, dan sigma menyimpan
nilai simpangan baku."
  )
  output$plotdNormal <- renderPlotly({
    dNorm.sample <- temp[input$dNorm_batasbawah:input$dNorm_batasatas]
    temp.mean <- input$dNorm_mean
    temp.sd <- input$dNorm_sd
    temp.dNorm <- dnorm(dNorm.sample, mean = temp.mean, sd = temp.sd)
    temp.df <- data.frame("Temperature" = dNorm.sample, "Density" = temp.dNorm)
    plotdNorm <- ggplot(temp.df, aes(x = Temperature, y = Density)) +
      geom_point(col = "maroon") +
      labs(title = "DENSITY FUNCTION OF NORMAL DISTRIBUTION", x = "TEMPERATURE", y = "DENSITY")+
      theme_minimal()
    ggplotly(plotdNorm)
  })
  output$histdNormal <- renderPlot({
    dNorm.sample <- temp[input$dNorm_batasbawah:input$dNorm_batasatas]

```

```

temp.mean <- input$dNorm_mean
temp.sd <- input$dNorm_sd
temp.dNorm <- dnorm(dNorm.sample, mean = temp.mean, sd = temp.sd)
histdNorm <- hist(temp.dNorm,
  main = "DENSITY FUNCTION OF NORMAL DISTRIBUTION",
  xlab = "Data Temperature Jawa Timur",
  col = brewer.pal(3, "PuBuGn"),
  border = "white")
})
output$dNorm_hasil <- renderDataTable({
  dNorm.sample <- temp[input$dNorm_batasbawah:input$dNorm_batasatas]
  temp.mean <- input$dNorm_mean
  temp.sd <- input$dNorm_sd
  temp.dNorm <- dnorm(dNorm.sample, mean = temp.mean, sd = temp.sd)
  dNorm_tabel <- data.frame(dNorm.sample, temp.dNorm)
  colnames(dNorm_tabel) <- c("TEMPERATURE SAMPLE", "PROBABILITAS HASIL D - NORM")
  expr = dNorm_tabel
})

# =====DISTRIBUSI UNIFORM=====
output$noteUniform <- renderText(
  "Dalam pengujian distribusi uniform ini data yang dibangkitkan sesuai dengan dataset, yaitu Data
  Temperature Maksimal Provinsi Jawa Timur Tahun 2020.
  Data tersebut akan dicetak ulang sebagai sample pengujian. Banyaknya sample dapat disesuaikan
  dengan input dari user, namun sample awal akan diset sebanyak
  1000 kali dengan perulangan. "
)
output$penjelasanUniform <- renderText(
  "Distribusi peluang uniform adalah distribusi yang mempunyai probabilitas yang sama pada setiap
  kejadian, tidak dikategorikan,
  dan ruang sampelnya tidak dibatasi."
)
output$histUniform <- renderPlot({
  generateRandom <- sample(temp, size = input$unif_input, replace = TRUE)
  generateRandom <- generateRandom/input$unif_input
  minimal <- min(generateRandom)
  maksimal <- max(generateRandom)
  histUniform <- hist(generateRandom,
    main = "PEMBANGKITAN DISTRIBUSI KONTINYU UNIFORM",
    xlab = "Variable Random Dari Data Temperature Jawa Timur",
    xlim = c(minimal, maksimal),
    col = brewer.pal(3, "Pastel2"),
    border = "white")
})
output$unif_hasil <- renderDataTable({
  generateRandom <- sample(temp, size = input$unif_input, replace = TRUE)
  hitung <- generateRandom/input$unif_input
  unif_tabel <- data.frame(generateRandom, hitung)
  colnames(unif_tabel) <- c("TEMPERATURE SAMPLE", "PROBABILITAS HASIL DISTRIBUSI UNIFORM")
  expr=unif_tabel
})

# =====HOME=====
output$temperature <- renderDataTable(dataku)

output$plotDesember <- renderPlotly({

```

```

diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = DESEMBER)) +
  geom_line(col = "navy") +
  geom_point(col = "pink") +
  labs(title = "TEMPERATURE MAKSIMUM BULAN DESEMBER", x = "TANGGAL", y = "TEMPERATURE")+
  theme_minimal()
ggplotly(diagram)
})

output$plotNovember <- renderPlotly({
  diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = NOVEMBER)) +
    geom_line(col = "navy") +
    geom_point(col = "pink") +
    labs(title = "TEMPERATURE MAKSIMUM BULAN NOVEMBER", x = "TANGGAL", y = "TEMPERATURE")+
    theme_minimal()
  ggplotly(diagram)
})

output$plotOktober<- renderPlotly({
  diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = OKTOBER)) +
    geom_line(col = "skyblue") +
    geom_point(col = "pink") +
    labs(title = "TEMPERATURE MAKSIMUM BULAN OKTOBER", x = "TANGGAL", y = "TEMPERATURE")+
    theme_minimal()
  ggplotly(diagram)
})

output$plotSeptember <- renderPlotly({
  diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = SEPTEMBER)) +
    geom_line(col = "navy") +
    geom_point(col = "pink") +
    labs(title = "TEMPERATURE MAKSIMUM SEPTEMBER", x = "TANGGAL", y = "TEMPERATURE")+
    theme_minimal()
  ggplotly(diagram)
})

output$plotAgustus<- renderPlotly({
  diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = AGUSTUS)) +
    geom_line(col = "skyblue") +
    geom_point(col = "pink") +
    labs(title = "TEMPERATURE MAKSIMUM BULAN AGUSTUS", x = "TANGGAL", y = "TEMPERATURE")+
    theme_minimal()
  ggplotly(diagram)
})

output$plotJuli <- renderPlotly({
  diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = JULI)) +
    geom_line(col = "navy") +
    geom_point(col = "pink") +
    labs(title = "TEMPERATURE MAKSIMUM BULAN JULI", x = "TANGGAL", y = "TEMPERATURE")+
    theme_minimal()
  ggplotly(diagram)
})

output$plotJuni<- renderPlotly({
  diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = JUNI)) +
    geom_line(col = "skyblue") +
    geom_point(col = "pink") +

```

```

      labs(title = "TEMPERATURE MAKSIMUM BULAN JUNI", x = "TANGGAL", y = "TEMPERATURE")+
      theme_minimal()
    ggplotly(diagram)
  })

  output$plotMei <- renderPlotly({
    diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = MEI)) +
      geom_line(col = "skyblue") +
      geom_point(col = "pink") +
      labs(title = "TEMPERATURE MAKSIMUM BULAN MEI", x = "TANGGAL", y = "TEMPERATURE")+
      theme_minimal()
    ggplotly(diagram)
  })

  output$plotApril<- renderPlotly({
    diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = APRIL)) +
      geom_line(col = "navy") +
      geom_point(col = "pink") +
      labs(title = "TEMPERATURE MAKSIMUM BULAN APRIL", x = "TANGGAL", y = "TEMPERATURE")+
      theme_minimal()
    ggplotly(diagram)
  })

  output$plotMaret <- renderPlotly({
    diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = MARET)) +
      geom_line(col = "skyblue") +
      geom_point(col = "pink") +
      labs(title = "TEMPERATURE MAKSIMUM BULAN MARET", x = "TANGGAL", y = "TEMPERATURE")+
      theme_minimal()
    ggplotly(diagram)
  })

  output$plotFebruari<- renderPlotly({
    diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = FEBRUARI)) +
      geom_line(col = "navy") +
      geom_point(col = "pink") +
      labs(title = "TEMPERATURE MAKSIMUM BULAN FEBRUARI", x = "TANGGAL", y = "TEMPERATURE")+
      theme_minimal()
    ggplotly(diagram)
  })

  output$plotJanuari <- renderPlotly({
    diagram <- ggplot(data = dataku, aes(x = TANGGAL, y = JANUARI)) +
      geom_line(col = "skyblue") +
      geom_point(col = "pink") +
      labs(title = "TEMPERATURE MAKSIMUM BULAN JANUARI", x = "TANGGAL", y = "TEMPERATURE")+
      theme_minimal()
    ggplotly(diagram)
  })

} # tutup kurung

```

❖ EVALUASI

Pada program berbasis R – Shiny yang telah dibuat masih terlihat banyak kekurangan, baik dari segi visualisasi ataupun keakuratan hasil. Oleh karena itu, program ini masih bisa untuk diperbaiki lagi. Salah satunya dengan lebih menganalisis hasil pengujian agar visualisasi data yang ditampilkan lebih akurat. Selain itu juga dapat dilakukan perbaikan dalam fitur aplikasi dan juga tata letaknya agar lebih terlihat menarik.

❖ KESIMPULAN

Pada program R sendiri, pengujian distribusi kontinu terbilang cukup mudah karena sudah terdapat fungsi yang tersedia untuk melakukan perhitungan dan beberapa tambahan parameter untuk menunjang perhitungan. Berikut ringkasan beberapa jenis distribusi kontinu beserta sintaks yang digunakan dalam pemrograman R.

- a. Distribusi Uniform : generate random number atau `rUnif(n, min, max)`
- b. Distribusi Normal
 - Density Function : `dnorm (x, mean, sd)`
 - Cumulative Density Function : `pnorm (x, mean, sd)`
 - Quantile Function : `qnorm (p, mean, sd)`
 - Random Sampling Distribution : `rnorm (n, mean, sd)`
- c. Distribusi Eksponensial
 - Density Function : `dexp(x_dexp, rate)`
 - Cumulative Density Function : `pexp(x_pexp, rate)`
 - Quantile Function : `qexp(x_qexp, rate)`
 - Random Sampling Distribution : `rexp(N, rate)`
- d. Distribusi Weibull : `dweibull(x, shape, scale, log)`