library(shiny)

library(readxl)

# UI

ui <- fluidPage(

tags$head(

tags$style(HTML("

.output-table {

font-family: Arial, sans-serif;

font-size: 12px;

border-collapse: collapse;

width: 100%;

}

.output-table th, .output-table td {

border: 1px solid #ddd;

padding: 8px;

}

.output-table th {

background-color: #f2f2f2;

font-weight: bold;

}

.output-table tr:nth-child(even) {

background-color: #f9f9f9;

}

.output-table tr:hover {

background-color: #ddd;

}

.output-box {

border: 1px solid #ddd;

padding: 10px;

margin-bottom: 10px;

}

"))

),

div(style = "text-align: center;", class = "output-box", HTML("

<h1>Aplikasi Peramalan Data Time Series</h1>

<h1>Menggunakan Metode Fuzzy Time Series Chen</h1>

<br>

<br>

<p>Inventor:</p>

<p>Puspita Kartikasari, S.Si., M.Si.</p>

<p>Deby Fakhriyana, S.Si., M.Si.</p>

<p>Marselinus Tolhas Gratias Lumbanbatu</p>

<br>

<br>

<p>DEPARTEMEN STATISTIKA</p>

<p>FAKULTAS SAINS DAN MATEMATIKA</p>

<p>UNIVERSITAS DIPONEGORO</p>

<p>TAHUN 2023</p>

")),

sidebarLayout(

sidebarPanel(

fileInput("file", "Unggah file Excel:", accept = ".xlsx"),

numericInput("D1", "Masukkan nilai D1:", value = 0),

numericInput("D2", "Masukkan nilai D2:", value = 0),

actionButton("hitungButton", "Hitung")

),

mainPanel(

tabsetPanel(

tabPanel("Data", div(class = "output-box", tableOutput("dataOutput"))),

tabPanel("Interval", div(class = "output-box", tableOutput("intervalOutput"))),

tabPanel("FLR", div(class = "output-box", tableOutput("flrOutput"))),

tabPanel("FLRG", div(class = "output-box", tableOutput("flrgOutput"))),

tabPanel("Ramalan", div(class = "output-box", verbatimTextOutput("ramalOutput"))

)

)

)

)

)

# Fungsi Fuzzy Time Series

interval\_fuzzy <- function(data, D1 = 0, D2 = 0) {

Umin <- min(data) - D1

Umax <- max(data) + D2

n <- round(1 + 3.322 \* log10(length(data)))

l <- (Umax - Umin) / n

intervals <- data.frame(mins = 0, maxs = 0)

intervals[1, 1] <- Umin

intervals[1, 2] <- Umin + l

for (i in 2:n) {

intervals[i, 1] <- intervals[i - 1, 2]

intervals[i, 2] <- intervals[i, 1] + l

}

return(intervals)

}

# Fungsi Fuzzy Chen

fuzzy\_chen <- function(data, interval) {

m <- as.vector(rowMeans(interval))

fuzifikasi <- c()

for (i in 1:length(data)) {

for (j in 1:nrow(interval)) {

if (i != which.max(data)) {

if (data[i] >= interval[j, 1] & data[i] < interval[j, 2]) {

fuzifikasi[i] <- j

break

}

} else {

if (data[i] >= interval[j, 1] & data[i] <= interval[j, 2]) {

fuzifikasi[i] <- j

break

}

}

}

}

flr <- data.frame(current\_state = 0, next\_state = 0)

for (i in 1:(length(fuzifikasi) - 1)) {

flr[i, ] <- c(fuzifikasi[i], fuzifikasi[i + 1])

}

flr[length(fuzifikasi), ] <- c(fuzifikasi[length(fuzifikasi)], 0)

flrg <- list()

for (i in 1:nrow(interval)) {

flrgi <- c()

for (j in 1:(length(data) - 1)) {

if (flr[j, 1] == i) {

flrgi <- c(flrgi, flr[j, 2])

}

}

flrg[[i]] <- flrgi

}

uni <- list()

for (i in 1:nrow(interval)) {

y <- flrg[[i]]

r <- unique(y)

uni[[i]] <- r

}

jum <- list()

for (i in 1:nrow(interval)) {

jums <- c()

for (j in 1:length(uni[[i]])) {

jums <- c(jums, m[uni[[i]][j]])

}

jum[[i]] <- jums

}

meanpred <- lapply(jum, mean)

prediksi <- c()

for (i in 1:(length(data) - 1)) {

pred <- meanpred[[fuzifikasi[i]]]

prediksi <- c(prediksi, pred)

}

ramal <- prediksi[length(prediksi)]

return(list(prediksi = prediksi, meanpred = meanpred, flr = flr, flrg = flrg, ramal = ramal))

}

# Server

server <- function(input, output) {

data <- reactive({

req(input$file)

df <- read\_excel(input$file$datapath)

df <- df[2:nrow(df), ] # Menghilangkan baris pertama

df$No <- seq\_len(nrow(df)) # Menambahkan kolom nomor urut

df$Data <- df$penutupan # Menambahkan kolom "Data"

df

})

interval <- reactive({

req(data())

interval\_fuzzy(data()$Data, D1 = input$D1, D2 = input$D2)

})

result <- eventReactive(input$hitungButton, {

req(data(), interval())

fuzzy\_chen(data()$Data, interval())

})

output$dataOutput <- renderTable({

df <- data()

df <- df[, c("No", "Data")]

colnames(df) <- c("No", "Data")

head(df, n = 15) # Menampilkan hanya 15 baris pertama

}, class = "output-table")

output$intervalOutput <- renderTable({

interval()

}, class = "output-table")

output$flrOutput <- renderTable({

flr <- result()$flr

flr$current\_state <- paste0("A", flr$current\_state)

flr$next\_state <- paste0("A", flr$next\_state)

flr

}, class = "output-table")

output$flrgOutput <- renderTable({

flrg <- result()$flrg

flrg <- lapply(flrg, function(x) paste0("A", x))

flrg\_df <- data.frame(

current\_state = rep(seq\_along(flrg), lengths(flrg)),

next\_state = unlist(flrg)

)

flrg\_df

}, class = "output-table")

output$ramalOutput <- renderPrint({

result()$ramal

})

}

# Run the application

shinyApp(ui = ui, server = server)