

# Prediksi Gempa Bumi di Indonesia Menggunakan R-Shiny

Yova Rezky Amanda<sup>1</sup>, Mumtazah Nurul 'Aini<sup>1</sup>, Melliyyuma Miyoze<sup>1</sup>,  
dan Dwi Oktavianto Wahyu Nugroho<sup>2</sup>

<sup>1</sup>Departemen Statistika Bisnis, Institut Teknologi Sepuluh Nopember (ITS)

<sup>2</sup>Departemen Teknik Instrumentasi, Institut Teknologi Sepuluh Nopember (ITS)

*e-mail:* oktavianto@instrum-eng.its.ac.id

**Abstrak**—Gempa bumi bisa terjadi akibat aktivitas vulkanis atau tektonik. Gempa bumi dalam skala besar bisa mengakibatkan kerusakan dan bahaya bagi manusia. Hal tersebut menyebabkan perlunya mengetahui info terkini terkait gempa bumi yang terjadi. Dengan mengetahui info terkini terkait gempa bumi, masyarakat akan lebih waspada terhadap kemungkinan gempa yang terjadi. Kini sudah banyak aplikasi yang memberikan info gempa bumi secara akurat. Aplikasi info gempa bumi ini menyajikan informasi terkait gempa yang terjadi secara real-time seiring bertumbuhnya kebutuhan aplikasi sistem informasi yang terpusat. Untuk dapat memprediksi sebuah gempa yang akan datang, ilmu yang bisa digunakan adalah peramalan. Metode peramalan dalam statistika (*forecasting*) adalah seni dan ilmu untuk memperkirakan kejadian di masa depan. Hal ini dapat dilakukan dengan melibatkan pengambilan data historis dan memroyeksikannya ke masa mendatang dengan suatu bentuk model matematis. Selain itu, bisa juga merupakan prediksi intuisi yang bersifat subjektif. Fungsi peramalan atau *forecasting* terlihat pada saat pengambilan keputusan. Keputusan yang baik adalah keputusan yang didasarkan atas pertimbangan apa yang akan terjadi pada waktu keputusan itu dilaksanakan. Untuk dapat melakukan peramalan, salah satu software yang bisa digunakan adalah R-Studio. R-Studio merupakan suatu *Integrated Development Environment (IDE)* untuk R. Software R adalah suatu sistem untuk melakukan analisis statistik dan grafik. R Studio merupakan software statistika yang populer, lengkap, dan berlisensi gratis sehingga peneliti dapat terhindar dari tuntutan hukum saat mengolah dan menganalisis data. Software ini memiliki visualisasi grafik yang lebih canggih dibandingkan software statistika lainnya. Selain itu, R Studio dapat digunakan dalam pembuatan aplikasi berbasis desktop maupun website. R-Shiny merupakan package dalam R-Studio yang berfungsi untuk membuat tampilan berupa web yang isinya terdapat diagram ataupun chart mengenai data yang telah diolah sebelumnya. Shiny adalah paket R *opensource* yang menyediakan kerangka web yang elegan dan kuat untuk membangun aplikasi web menggunakan R. Shiny dapat membantu mengubah analisis menjadi aplikasi web interaktif tanpa memerlukan pengetahuan HTML, CSS atau JavaScript. Penelitian ini bertujuan mengetahui cara membangun sistem untuk memprediksi gempa dengan cara merancang dan menjalankan web atau aplikasi deteksi gempa bumi di Indonesia pada tahap berikutnya. Manfaat yang dapat diperoleh pada penelitian kali ini bagi praktisi adalah dapat lebih memahami cara membangun sistem untuk memprediksi gempa menggunakan software dan bagi pembaca adalah dapat mengetahui informasi terkini mengenai prediksi gempa yang akan terjadi di masa mendatang.

**Kata Kunci**—Gempa Bumi, Prediksi, R-Shiny.

## I. PENDAHULUAN

**G**EMPA bumi bisa terjadi akibat aktivitas vulkanis atau tektonik. Gempa bumi dalam skala besar bisa mengakibatkan kerusakan dan bahaya bagi manusia. Ini

pentingnya mengetahui info terkini terkait gempa bumi yang terjadi. Mengetahui info gempa terkini bisa membuatmu lebih waspada terhadap kemungkinan bencana yang bisa terjadi. Kini sudah banyak aplikasi yang memberikan info gempa bumi secara akurat. Aplikasi info gempa bumi ini menyajikan informasi terkait gempa yang terjadi secara *real-time*. Seiring bertumbuhnya kebutuhan aplikasi sistem informasi yang terpusat. Untuk dapat memprediksi sebuah gempa yang akan datang, ilmu yang bisa digunakan adalah peramalan.

Peramalan (*forecasting*) adalah seni dan ilmu untuk memperkirakan kejadian di masa depan [1]. Hal ini dapat dilakukan dengan melibatkan pengambilan data historis dan memroyeksikannya ke masa mendatang dengan suatu bentuk model matematis. Selain itu, bisa juga merupakan prediksi intuisi yang bersifat subjektif. Fungsi peramalan atau *forecasting* terlihat pada saat pengambilan keputusan. Keputusan yang baik adalah keputusan yang didasarkan atas pertimbangan apa yang akan terjadi pada waktu keputusan itu dilaksanakan.

Untuk dapat melakukan peramalan, salah satu *software* yang bisa digunakan adalah R-Studio. R-Studio merupakan suatu *Integrated Development Environment (IDE)* untuk R. Software R adalah suatu sistem untuk melakukan analisis statistik dan grafik[1]. R Studio merupakan software statistika yang populer, lengkap, dan berlisensi gratis sehingga peneliti dapat terhindar dari tuntutan hukum saat mengolah dan menganalisis data. Software ini memiliki visualisasi grafik yang lebih canggih dibandingkan software statistika lainnya. Selain itu, R Studio juga dapat digunakan dalam pembuatan aplikasi berbasis desktop maupun website. R-Shiny merupakan package dalam R-Studio yang berfungsi untuk membuat tampilan berupa web yang isinya terdapat diagram ataupun chart mengenai data yang telah diolah sebelumnya. Shiny adalah paket R *opensource* yang menyediakan kerangka web yang elegan dan kuat untuk membangun aplikasi web menggunakan R. Shiny dapat membantu mengubah analisis menjadi aplikasi web interaktif tanpa memerlukan pengetahuan HTML, CSS atau JavaScript[1].

Penelitian ini bertujuan mengetahui cara membangun sistem untuk memprediksi gempa dengan cara merancang dan menjalankan web atau aplikasi deteksi gempa bumi di Indonesia pada tahap berikutnya. Manfaat yang dapat diperoleh pada praktikum kali ini bagi praktisi adalah dapat lebih memahami cara membangun sistem untuk memprediksi gempa menggunakan *software* dan bagi pembaca adalah dapat mengetahui informasi terkini mengenai prediksi gempa yang akan terjadi di masa mendatang. *Software* ini membantu pengguna untuk lebih waspada terhadap gempa.

## II. TINJAUAN PUSTAKA

### A. Gempa Bumi

Gempa bumi adalah peristiwa bergetarnya bumi akibat pelepasan energi dari dalam perut bumi secara tiba-tiba, sehingga menciptakan gelombang seismik, yang ditandai dengan patahnya lapisan batuan pada kerak bumi [1]. Akumulasi energi penyebab terjadinya gempa bumi dihasilkan dari pergerakan lempeng-lempeng tektonik. Energi yang dihasilkan dipancarkan kesegala arah berupa gelombang gempa bumi sehingga efeknya dapat dirasakan sampai ke permukaan bumi [1]. Gempa bumi juga dapat diakibatkan aktifitas gunung berapi, tanah longsor dan meteor yang menumbuk bumi. Menurut teori lempeng tektonik, kerak bumi terpecah-pecah menjadi beberapa bagian yang disebut lempeng. Lempeng-lempeng tersebut bergerak dengan arah dan kecepatan berbeda. Pergerakan lempeng ini disebabkan oleh arus konveksi. Lapisan atas bumi terdiri dari lithosfer dan asthenosfer. Lithosfer mempunyai densitas yang lebih besar, mudah patah, dan bersifat kaku. Asthenosfer mempunyai densitas yang lebih kecil dibandingkan lithosfer, bersuhu tinggi dan kental. Akibat gerakan perputaran bumi yang terus-menerus menimbulkan arus pada asthenosfer yang bersuhu tinggi. Arus ini disebut arus konveksi, yang bergerak dari tekanan tinggi ke tempat yang bertekanan rendah. Gerakan asthenosfer akan menggerakkan lithosfer yang mengapung di atasnya, akibatnya lithosfer yang berupa lempeng-lempeng akan bergerak.

Gempa bumi dapat diklasifikasikan berdasarkan sumber penyebab terjadinya [1].

1. Gempa bumi runtuh (Collapse Earthquake) adalah gempa bumi yang disebabkan oleh runtuhnya lubang-lubang di dalam bumi, seperti gua, tambang dan sebagainya.
2. Gempa bumi vulkanik (Volcanic Earthquake) adalah gempa bumi yang berasal dari gerakan magma karena aktivitas gunung api.
3. Gempa bumi Tektonik (Tectonic Earthquake) adalah gempa bumi yang disebabkan oleh aktifitas sesar karena perlipatan kerak bumi, pembentukan pegunungan dan sebagainya. Gempa bumi tektonik ini merupakan gempa bumi yang signifikan terjadi di bumi secara menyeluruh.

### B. Longitude dan Latitude

*Latitude* disebut juga garis lintang. Garis lintang merupakan garis vertikal yang mengukur sudut antara suatu titik dengan garis katulistiwa. Titik di utara garis katulistiwa dinamakan lintang utara, sedangkan titik di selatan katulistiwa dinamakan lintang selatan. *Longitude* disebut juga garis bujur. Garis bujur yaitu garis horizontal yang mengukur sudut antara satu titik dengan titik nol bumi, yaitu Greenwich di London, Britania Raya yang merupakan titik 0 0 atau 3600 yang diterima secara internasional. Titik dibarat bujur 0 0 dinamakan bujur barat, sedangkan titik di timur 3600 dinamakan bujur timur [1].

### C. Kedalaman

Kedalaman sumber gempa bumi adalah jarak hiposenter dihitung tegak lurus dari permukaan bumi. Kedalaman dinyatakan oleh besaran jarak dalam satuan kilometer (km). kedalaman hiposenter adalah jarak hiposenter dihitung tegak

lurus dari permukaan bumi. Kedalaman dinyatakan oleh besaran jarak dalam satuan kolimeter [1].

### D. Magnitudo

Skala magnitudo berdasarkan pada beberapa asumsi sederhana, yaitu: a. Magnitudo adalah ukuran energi yang dilepaskan oleh batuan yang sebanding dengan kecepatan gerakan tanah, yaitu perbandingan amplitudo (A) dengan periode (T). b. Dua gempa dengan kekuatan yang berbeda dan direkam dengan geometri sumberpenerima yang sama maka kejadian yang lebih besar adalah yang akan menghasilkan amplitudo yang lebih besar. c. Penurunan amplitudo karena efek geometri dan atenuasi dapat diselesaikan secara statistik dengan fungsi kalibrasi  $F(\Delta, h)$ . d. Efek sumber seperti directivity dapat dikoreksi secara regional ( $Cr$ ), dan pengaruh lokal seperti struktur batuan lokal, topografi dan lain-lain dikoreksi dengan stasiun ( $Cs$ ) [1]. Berdasarkan asumsi-asumsi di atas, maka bentuk umum dari skala magnitudo dapat dituliskan pada Persamaan 1 [1].

$$M = \log AT + F(\Delta, h)Cs + Cr \quad (1)$$

di mana:

A: amplitudo ( $\mu\text{m}$ )

T: periode (detik)

F: koreksi jarak episenter dan kedalaman

h: kedalaman (*focal depth*)

$\Delta$ : jarak episenter

Cs: koreksi tempat stasiun

Cr: koreksi daerah sumber

### E. Peramalan

Peramalan (*forecasting*) adalah seni dan ilmu untuk memperkirakan kejadian di masa depan [1]. Hal ini dapat dilakukan dengan melibatkan pengambilan data historis dan memproyeksikannya ke masa mendatang dengan suatu bentuk model matematis. Selain itu, bisa juga merupakan prediksi intuisi yang bersifat subjektif. Fungsi peramalan atau forecasting terlihat pada saat pengambilan keputusan. Keputusan yang baik adalah keputusan yang didasarkan atas pertimbangan apa yang akan terjadi pada waktu keputusan itu dilaksanakan. Apabila kurang tepat ramalan yang kita susun, maka masalah peramalan juga merupakan masalah yang selalu kita hadapi [1]. Peramalan permintaan memiliki karakteristik tertentu yang berlaku secara umum. Karakteristik ini harus diperhatikan untuk menilai hasil suatu proses peramalan permintaan dan metode peramalan yang digunakan. Karakteristik peramalan yaitu faktor penyebab yang berlaku di masa lalu diasumsikan akan berlaku juga di masa yang akan datang, dan peramalan tak pernah sempurna, permintaan aktual selalu berbeda dengan permintaan yang diramalkan [1].

#### 1. Single Moving Average

*Moving Averages* (rata-rata bergerak) adalah metode peramalan perataan nilai dengan mengambil sekelompok nilai pengamatan yang kemudian dicari rata-ratanya, lalu menggunakan rata-rata tersebut sebagai ramalan untuk periode berikutnya [1]. Istilah rata-rata bergerak digunakan, karena setiap kali data observasi baru tersedia, maka angka rata-rata yang baru dihitung dan dipergunakan sebagai ramalan. Rata-rata bergerak tunggal (*Single Moving Average*) adalah suatu metode peramalan yang dilakukan dengan

Tabel 1.  
Variabel Penelitian

Variabel ke-	Keterangan	Skala
1	Waktu terjadinya gempa	Rasio
2	Latitude	Rasio
3	Longitude	Rasio
4	Depth	Rasio
5	Magnitudo (M)	Rasio
6	Region	Nominal

```
library(tidyquant)
library(forecast)
library(TTR)
library(smooth)
```

Gambar 1. Library Script Predict.R.

```
pastData <- data.frame(matrix(ncol = 7, nrow = 0))
x <- c("Date", "Time", "Longitude", "Latitude", "Depth",
      "Mag", "Region")
colnames(pastData) <- x
for (year in 2016:2021) {
  tempData = read.csv(paste("Data_Lengkap", ".csv", sep = ""))
  pastData <- rbind(pastData, tempData)
}
pastData <- na.omit(pastData)
pastData$Date <- as.Date(pastData$Date, format =
"%m/%d/%Y")
pastData$DT <- as.POSIXct(paste(pastData$Date,
pastData$Time), format = "%Y-%m-%d %H:%M:%OS")
startDateTime <- strptime("2016-2-1 00:00:00", format = "%Y-
%m-%d %H:%M:%OS")
endDateTime <- strptime("2021-2-28 00:00:00", format = "%Y-
%m-%d %H:%M:%OS")
minMag <- 3
maxMag <- 9
toPlot <- pastData[pastData$DT < endDateTime & pastData$DT >
startDateTime & pastData$Mag < maxMag & pastData$Mag >
minMag,]
View(toPlot$Date)
freqData <- cbind.data.frame(toPlot$Date, toPlot$Region, 0, 1)
colnames(freqData) <- c("Date", "Region", "Frequency",
"RegionID")

View(freqData)
```

Gambar 2. Impor Data.

mengambil sekelompok nilai pengamatan, mencari nilai rata-rata tersebut sebagai ramalan untuk periode yang akan datang. Metode *Single Moving Average* mempunyai karakteristik khusus yaitu: (a) untuk menentukan ramalan pada periode yang akan datang memerlukan data historis selama jangka waktu tertentu. Misalnya, dengan 3 bulan *moving average*, maka ramalan bulan ke 5 baru dibuat setelah bulan ke 4 selesai/berakhir. Jika bulan *moving average* bulan ke 7 baru bisa dibuat setelah bulan ke 6 berakhir; (b) semakin panjang jangka waktu *moving average*, efek pelicinan semakin terlihat dalam ramalan atau menghasilkan *moving average* yang semakin halus.

## 2. Mean Square Error

Mean Squared Error (MSE) adalah metode lain untuk mengevaluasi metode peramalan. Masing-masing kesalahan atau sisa dikuadratkan. Kemudian dijumlahkan dan ditambahkan dengan jumlah observasi. Pendekatan ini mengatur kesalahan peramalan yang besar karena kesalahan-kesalahan itu dikuadratkan. Metode itu menghasilkan

```
# Get list of regions
regions <- unique(pastData$Region)
# Map each region from CSV to region to be outputted
regCSV = read.csv("RegionMapping.csv", sep = ",")
for (i in 1:dim(regCSV)[1]) {
  row <- as.numeric(regCSV[i, 2:5])
  for (j in 1:length(row)) {
    if (is.na(row[j])) break
    freqData[freqData$Region %in% regions[row[j]], 4] <- i
  }
}
freqData <- freqData[freqData$RegionID != 0, colnames(freqData)
! = "Region"]

# Convert date from Date format to ts format (ex: 2008.173)
getTSDate <- function(dateTime) {
  return(year(dateTime) + yday(dateTime) / 364)
}
# Convert date from ts format to Date format (ex: 2008-03-04)
getDateFromTS <- function(dateTime) {
  year <- floor(dateTime)
  noOfWeeks <- floor((dateTime - year) * 365 / 7)
  dt <- as.Date(paste(year, 1, 1, sep = "-"), format = "%Y-%m-%d")
  dt <- dt + weeks(noOfWeeks)
  return(as.Date(dt))
}
getDateFromYearMon <- function(dateTime) {
  return(as.Date(paste(dateTime[1], dateTIme[2], "1", sep = "-"),
format = "%Y-%m-%d"))
}

# Function to return ts object containing frequency data of region
getRegionData <- function(regionID) {
  regData <- freqData[freqData$RegionID == regionID,
colnames(freqData) != "RegionID"]

  # Padding data
  time.min <- freqData$Date[1]
  time.max <- freqData$Date[dim(freqData)[1]]
  allDates <- seq(time.min, time.max, by = "day")
  allDates.frame <- data.frame(list(Date = allDates))
  regData <- merge(allDates.frame, regData, all = TRUE)
  regData$Frequency[which(is.na(regData$Frequency))] <- 0
}
```

Gambar 3. Pemetaan Wilayah.

kesalahan-kesalahan sedang yang kemungkinan lebih baik untuk kesalahan kecil, tetapi kadang menghasilkan perbedaan yang besar[1].

## F. R-Studio

R-Studio merupakan suatu *Integrated Development Environment* (IDE) untuk R. *Software* R adalah suatu sistem untuk melakukan analisis statistik dan grafik[1]. R Studio merupakan software statistika yang populer, lengkap, dan berlisensi gratis sehingga peneliti dapat terhindar dari tuntutan hukum saat mengolah dan menganalisis data. Software ini memiliki visualisasi grafik yang lebih canggih dibandingkan software statistika lainnya. Selain itu, R Studio juga dapat digunakan dalam pembuatan aplikasi berbasis desktop maupun website.

## G. R-Shiny

R-Shiny merupakan *package* dalam R-Studio yang berfungsi untuk membuat tampilan berupa web yang isinya terdapat diagram ataupun chart mengenai data yang telah diolah sebelumnya. *Shiny* adalah paket R *opensource* yang menyediakan kerangka web yang elegan dan kuat untuk membangun aplikasi web menggunakan R. Shiny dapat membantu mengubah analisis menjadi aplikasi web interaktif tanpa memerlukan pengetahuan HTML, CSS atau JavaScript[1]. Komponen dalam R-Shiny ada 2 kelompok

```

# Forecast time series with moving-average of order 1
# Store all forecasts in 1 df for cross-validation
while (dt <= finalDateTime) {
  ts <- ts(as.numeric(regData.all), start = regData.start, end =
tempEnd, frequency = 52)
  fd <- rbind(fd, data.frame(forecast(ma(ts, order = 1),
weekRange)$mean[1:weekRange]))
  fd.upper <- rbind(fd.upper, data.frame(forecast(ma(ts, order
= 1), weekRange)$upper[1:weekRange, 1]))
  fd.residuals <- forecast(ts, weekRange)$residuals
  dt <- as.Date(paste(tempEnd[1], tempEnd[2], tempEnd[3],
sep = "-"), format = "%Y-%m-%d")
  dt <- dt + weeks(weekRange)
  tempEnd <- c(year(dt), month(dt), day(dt))
}
names(fd) <- "Frequency"
names(fd.upper) <- "Frequency"
fd <- ts(as.numeric(fd)$Frequency), start = regData.end,
frequency = 52)
fd.upper <- ts(as.numeric(fd.upper)$Frequency), start =
regData.end, frequency = 52)
# Create plot and print accuracy
png(file = path, width = 800, height = 600, bg = "transparent")
#regData.weekly <- tsclean(regData.weekly)
regData.start <- c(year(start(regData.weekly)),
month(start(regData.weekly)), day(start(regData.weekly)))
regData.end <- c(year(end(regData.weekly)),
month(end(regData.weekly)), day(end(regData.weekly)))
regData.all <- ts(as.numeric(regData.weekly), start =
regData.start, end = regData.end, frequency = 52)
return(regData.all)
}
# Function to return 1-dimensional df containing forecast data
based on historical data
getForecast <- function(regData.all, noOfWeeks) {
  weekRange = 26
  # Get start date
  #dt <- as.Date(as.yearmon(time(regData.all)[1]))
  dt <- getDateFromYearMon(start(regData.all))
  regData.start <- c(year(dt), month(dt), day(dt))
  # Get first end date (end date is to be iterated)
  dt <- dt + weeks(weekRange)
  regData.end <- c(year(dt), month(dt), day(dt))
  tempEnd <- regData.end
  finalDateTime <- endDateTime + weeks(noOfWeeks)
  fd <- c()
  fd.upper <- c(
  plot(fd)
  lines(regData.all, col = "red")
  dev.off()

```

Gambar 4. Prediksi Frekuensi Gempa.

```

#acc <- accuracy(fd, ma(regData.all, order = 1))
#print(acc)
#hist(fd.residuals)
# Cut the forecast data to appropriate range
cutStartDate<-getDateFromYearMon(start(regData.all)) +
weeks(weekRange) * 4
cutEndDate <- getDateFromYearMon(end(regData.all)) +
weeks(noOfWeeks)
finalFd <- window(fd, getTSDate(cutStartDate),
getTSDate(cutEndDate))
# Change negative numbers to 0
for (i in 1:length(finalFd)) {
  finalFd[i] = max(finalFd[i], 0)
}
return(finalFd)
}
# Get forecast for all regions by iteration
for (i in 1:dim(regCSV)[1]) {
  # Path to save plots
  path <- file.path(paste("plot_", i, ".png", sep = ""))

  regDf <- getRegionData(i)
  cleanedRegDf <- tsclean(regDf)
  fd <- getForecast(cleanedRegDf, 52)
  regAcc <- accuracy(fd, cleanedRegDf)
  if (i == 1) {
    pastDf <- data.frame(matrix(ncol = 13, nrow =
nrow(data.frame(regDf))))
    forecastDf <- data.frame(matrix(ncol = 13, nrow =
nrow(data.frame(fd))))
    acc <- data.frame(matrix(ncol = 3, nrow = 1))
    x <- c("Time", paste("R", 1:12, sep = ""))
    colnames(pastDf) <- x
    colnames(forecastDf) <- x
    colnames(acc) <- c("ME", "RMSE", "MAE")

    pastDf[1] <- round(time(regDf), digits = 3)
    forecastDf[1] <- round(time(fd), digits = 3)
  }

  pastDf[i + 1] <- data.frame(regDf)
  forecastDf[i + 1] <- data.frame(fd)
  acc[i,] <- regAcc[1:3]
}

# Save both original and forecast df to files
saveRDS(pastDf, file = "FreqPast.rds")
saveRDS(forecastDf, file = "FreqForecast.rds")
saveRDS(acc, file = "FreqAccuracy.rds")

```

```

# Shape file edited from
https://geonode.wfp.org/layers/geonode%3Aarchive
curRegions <- geojsonio::geojson_read("Regions.json",
what = "sp")
# Load files from previous script as df
pastDf <- readRDS(file = "FreqPast.rds")
forecastDf <- readRDS(file = "FreqForecast.rds")
accDf <- readRDS(file = "FreqAccuracy.rds")
View(forecastDf)
# Initialise frequency column
curRegions@data <-
as.data.frame(cbind(curRegions@data, rep(0, 12)))

```

Gambar 5. Proses Inisialisasi Prediksi.

```

names(curRegions@data)[dim(curRegions@data)[2]] <-
"Frequency"
# Year selection for historical and frequency data
yearSelection <- list()
startYear <- floor(pastDf[1,]$Time)
endYear <- floor(pastDf[length(pastDf$Time),]$Time)
yearSelection[[1]] <- c(startYear:endYear)
startYear <- floor(forecastDf[1,]$Time)
endYear <-
floor(forecastDf[length(forecastDf$Time),]$Time)
yearSelection[[2]] <- c(startYear:endYear)

```

```

library(shiny)
library(shinydashboard)
library(leaflet)
library(geojsonio)

```

Gambar 6. Package yang Dibutuhkan.

```

# Initialise ID of region which is clicked
regID <- NULL
ui <- fluidPage(
  fluidRow(
    dashboardPage(
      dashboardHeader(
        title = "Earthquake Frequency",
        titleWidth = 250
      ),
      dashboardSidebar(
        selectInput("dataType", "Display", c("Historical",
"Forecast")),
        # Dynamic year selection input based on data type
        uiOutput("yearSelection"),
        # Dynamic week slider input based on year and
data type
        uiOutput("weekSlider"),
        br(),
        uiOutput("plotImage", align = "center"),
        htmlOutput("printAcc"),
        width = 250
      ),
      dashboardBody(
        # Make height of map fill screen
        tags$style(type = "text/css", "#map {height:
calc(100vh - 80px) !important;}"),
        leafletOutput("map")
      )
    )
  )
)
server <- function(input, output) {
  # Get available years from year selection list
  output$yearSelection <- renderUI({
    if (input$dataType == "Historical") id = 1
    else id = 2
    selectInput("year", "Year", yearSelection[[id], selected
= input$year)
  })
  # Get minimum and maximum week based on year and
data type
  output$weekSlider <- renderUI({
    if (length(input$year) > 0) {
      if (input$year == yearSelection[[1]][1] &&
input$dataType == "Historical") {
        minDT <- pastDf$Time[1]
        minWeek = round((minDT - floor(minDT)) * 52,
digits = 0)
        maxWeek = 51
        sliderInput("week", "Week", min = minWeek, max
= maxWeek, value = max(minWeek, input$week))
      }
      else if (input$year == yearSelection[[2]][1] &&
input$dataType == "Forecast") {
        minDT <- forecastDf$Time[1]
        minWeek = round((minDT - floor(minDT)) * 52,
digits = 0)
        maxWeek = 51
        sliderInput("week", "Week", min = minWeek, max
= maxWeek, value = max(minWeek, input$week))
      }
      else if (input$year ==
yearSelection[[1]][length(yearSelection[[1]])] &&
input$dataType == "Historical") {
        maxDT <- pastDf$Time[length(pastDf$Time)]
        minWeek = 0
        maxWeek = round((maxDT - floor(maxDT)) * 52,
digits = 0)
        sliderInput("week", "Week", min = minWeek, max
= maxWeek, value = min(maxWeek, input$week))
      }
      else if (input$year ==
yearSelection[[2]][length(yearSelection[[2]])] &&

```

```

input$dataType == "Forecast") {
        maxDT <-
forecastDf$Time[length(forecastDf$Time)]
        minWeek = 0
        maxWeek = round((maxDT - floor(maxDT)) * 52,
digits = 0)
        sliderInput("week", "Week", min = minWeek, max
= maxWeek, value = min(maxWeek, input$week))
      }
      else {
        minWeek = 0
        maxWeek = 51
        sliderInput("week", "Week", min = minWeek, max
= maxWeek, value = input$week)
      }
    }
  })
  # Update earthquake frequency data to show in map from
year, week and data type selected
  updateFrequency <- function() {
    if (length(input$year) > 0 && length(input$week) > 0)
    {
      dateTime <- round(as.numeric(input$year) +
(as.numeric(input$week) * 7) / 364, digits = 3)
      if (input$dataType == "Historical") {
        curTimeData <- as.matrix(pastDf[pastDf$Time ==
dateTime, 2:dim(pastDf)[2]][1,])
      }
      else {
        curTimeData <-
as.matrix(forecastDf[forecastDf$Time == dateTime,
2:dim(forecastDf)[2]][1,])
      }
      return(curTimeData)
    }
    else
      return(NULL)
  }
  # Get ID of region which is clicked on map
  getRegID <- eventReactive(input$map_shape_click, {
    p <- input$map_shape_click
    regID <- which(curRegions@data$A1NAME == p$Id)
    return(regID)
  })
  #TODO - find directory of images
  output$plotImage <- renderUI({
    if (!is.null(getRegID())) {
      regID <- getRegID()
      img(src = paste('plot_', regID, '.png', sep = ""), align
= "center", width = 225)
    }
  })
  # Print error of region prediction
  output$printAcc <- renderUI({
    if (!is.null(getRegID())) {
      regID <- getRegID()
      regAcc <- accDf[regID,]
      regAcc <- format(regAcc, digits = 5)
      HTML(paste("<center><h4><b>",
curRegions@data$A1NAME[regID],
"</h4>ME:</b>", regAcc[1],
"<br/><b>RMSE:</b>", regAcc[2],
"<br/><b>MAE:</b>", regAcc[3], "</center>",
sep = ""))
    }
  })
  output$map <- renderLeaflet({
    # Check if update frequency function has updated with
new values (based on new selections)

    # If yes, override current frequency column with new
frequency
    df <- updateFrequency()
    if (!is.null(df)) {
      curRegions@data$Frequency <- df
    }
  })

```

Gambar 7. Pembuatan UI dan Server.

```

# Colours and labels
bins <- c(0, 2, 4, 6, 8, Inf)
pal <- colorBin("YlOrRd", domain = curRegions$Frequency,
bins = bins)
labels <- sprintf("<strong>%s</strong>",
curRegions$A1NAME) %>% lapply(htmltools::HTML)
# Choropleth map based on frequency data
m <- leaflet(curRegions) %>%
# Add maptiles
addProviderTiles(providers$CartoDB.Positron, group =
"Default Maptile") %>%
addProviderTiles(providers$CartoDB.DarkMatter, group =
"Dark Maptile") %>%
addProviderTiles(providers$Esri.WorldImagery, group =
"Satellite Maptile") %>%
addLayersControl(
baseGroups = c("Default Maptile", "Dark Maptile",
"Satellite Maptile"),
options = layersControlOptions(collapsed = FALSE))
%>%
# Set view to zoom on Indonesia
setView(118, -2, zoom = 5) %>%
# Draw region shapes on map based on geojson file
# Region colours based on frequency data
addPolygons(
fillColor = ~pal(Frequency),
weight = 2,
opacity = 0.8,
fillOpacity = 0.6,
#color = "#8b0000",
#color = "#ff7f00",
color = "#cc5500",
layerId = curRegions@data$A1NAME,
highlight = highlightOptions(
weight = 5,
fillOpacity = 0.8,
bringToFront = TRUE,
label = labels,
labelOptions = labelOptions(
style = list("font-weight" = "normal", padding =
"3px 8px"),
textsize = "15px",
direction = "auto")) %>%
addLegend(pal = pal, values = ~Frequency, opacity =
0.7,
title = NULL, position = "bottomright")
})
}

```

Gambar 7. Pembuatan UI dan Server (lanjutan).

yaitu *user interface* dan *server*. *User interface* berupa file HTML yang digunakan untuk memasukkan nilai input dan penyajian output.

Secara umum, komponen pada program *Shiny* dibedakan menjadi dua kelompok besar, yaitu *User Interface* (UI) dan *Server*. Untuk penjelasan lebih lengkapnya sebagai berikut:

### 1. User Interface (UI)

Bagian ini bermanfaat untuk: (a) panel kontrol adalah panel untuk mengontrol input berupa data, variabel, model, tergantung kompleksitas modul. Tampilan kontrol dapat berupa slider, radio button, check box, dan lain-lain; (b) pemasukan permintaan nilai input yaitu berupa data dengan berbagai jenis variabel yang diperlukan, pemilihan model, jenis dan kriteria uji statistika; (c) penyajian output yaitu terkait hasil atau uji. Hasil output dapat berupa grafik (histogram, diagram pencar, dan lain-lain), bentuk angka atau teks bisa berbentuk asli (verbatim) maupun dalam bentuk tabel, dan teks khusus bernotasi matematika dengan format latex; (d) untuk mengakomodasi berbagai jenis luaran tadi, shiny juga menyediakan berbagai format output seperti

plotoutput, textoutput, vebatimtextoutput, tableoutput, dan lain-lain. Bagian ini dapat disajikan pada file khusus ui.r, dan dapat juga disajikan penuh melalui file html, misalnya index.html.

### 2. Server

Bagian yang merupakan otak dari program yang bertugas melakukan simulasi, berbagai analisis data sesuai pilihan pengguna dan selanjutnya mengirim hasilnya ke bagian output. Bagian ini didukung oleh berbagai prosedur analisis data yang pada umumnya telah tersedia pada berbagai paket r. Bagian ini disimpan dalam file yang diberi nama server.

## III. METODOLOGI PENELITIAN

### A. Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yakni data yang diperoleh dari web *repogempa.bmkg.go.id*. Adapun waktu perolehan data, yaitu pada hari Minggu pada tanggal 28 Maret 2021 di Jawa Timur.

### B. Variabel Penelitian

Variabel yang digunakan untuk memprediksi frekuensi gempa bumi di Indonesia ditunjukkan pada Tabel 1.

### C. Langkah Analisis

Langkah analisis yang dilakukan pada penelitian ini dijelaskan sebagai berikut.

1. Mengimpor data Gempa Bumi di Indonesia selama 5 tahun terakhir yaitu Februari 2016 sampai Februari 2021 dalam bentuk excel.
2. Melakukan prediksi berdasarkan data gempa yang telah diambil, yaitu: (a) Melakukan pemetaan wilayah yang terdapat dalam data ke wilayah yang akan ditampilkan dalam pemetaan; (b) Mengelompokkan data gempa; (c) Prediksi dilakukan dengan memodelkan data menggunakan *Moving Average* (MA) orde 1 dikarenakan *Autocorrelation function* (ACF) dari data mempunyai nilai tidak signifikan setelah lag 1; (d) Menghitung error dari hasil prediksi; (e) Menyimpan hasil prediksi ke dalam file RDS.
3. Visualisasi data dilakukan dengan menggunakan aplikasi Shiny. Prediksi frekuensi gempa ditampilkan pada peta Indonesia yang terdapat dalam aplikasi.

## IV. HASIL DAN PEMBAHASAN

Pada bab kali ini akan membahas mengenai hasil proses tahapan meliputi impor data pada R, hingga melakukan visualisasi hasil prediksi gempa menggunakan Shiny.

### A. Menjalankan Script Predict.R untuk Melakukan Prediksi Berdasarkan Data Gempa Indonesia Tahun 2016 - 2021

Untuk menjalankan script predict.R diperlukan *import package* yang digunakan dalam R seperti pada Gambar 1. *Library* yang telah diimpor selanjutnya digunakan untuk pemetaan wilayah, mengelompokkan data gempa, memprediksi frekuensi gempa dengan metode *Single Moving Average*, hingga menghitung error dan hasil prediksi. Sebelum melakukan tahapan tersebut terlebih dahulu dilakukan impor data gempa Indonesia tahun 2016-2021 yang ada pada file excel dalam format csv ke dalam R.

```
# Create Shiny app
shinyApp(ui, server)
```

Gambar 8. *Syntax* Visualisasi *Shiny*.

Gambar 9. Visualisasi Prediksi Gempa.

Proses pada Gambar 2 menunjukkan proses mengenai impor data, pendefinisian kolom berdasarkan variabel, serta pendefinisian range waktu serta kekuatan gempa (Magnitudo) berdasarkan data yang didapatkan. Langkah selanjutnya adalah pemetaan wilayah dapat dilihat pada gambar 3.

Proses pada Gambar 3 merupakan proses pemetaan wilayah gempa yang selanjutnya hasil pemetaan wilayah tersebut akan digunakan untuk memprediksi frekuensi gempa menggunakan metode *Single Moving Average* dengan langkah yang disajikan pada Gambar 4.

Proses yang dilakukan pada Gambar 4 meliputi prediksi frekuensi gempa dengan metode *Single Moving Average*, menghitung error dan hasil prediksi berdasarkan data gempa Indonesia tahun 2016-2021, hingga menyimpan hasil prediksi kedalam file RDS. Proses selanjutnya adalah melakukan pembuatan UI dan Server sebelum melakukan visualisasi dengan menggunakan *Shiny*.

### B. Pembuatan UI dan Server

Pembuatan UI dan Server serta visualisasi prediksi frekuensi terjadinya gempa bumi menggunakan RStudio membutuhkan beberapa *package* yang harus diimpor seperti pada Gambar 6.

Gambar 6 menunjukkan package-package yang diperlukan dalam pembuatan aplikasi. Jika *package-package* tersebut belum *ter-install* dalam program RStudio, maka perlu diinstall terlebih dahulu dengan bantuan perintah *install.packages(nama package)*.

Selanjutnya, dilanjutkan dengan proses inisialisasi prediksi frekuensi gempa dan waktu terjadinya gempa dengan *syntax* seperti pada Gambar 5. Setelah melakukan proses inisialisasi prediksi, kemudian dilanjutkan dengan pembuatan UI dan Server dengan proses dan *syntax* yang ditunjukkan oleh Gambar 7.

Pembuatan UI dan Server seperti pada Gambar 7 digunakan untuk membuat panel kontrol untuk mengontrol input dan tampilan kontrol berupa *slider*, *checkbox*, dan lain-lain, memasukkan nilai input yaitu berupa data dengan berbagai jenis variabel yang diperlukan, pemilihan model, jenis dan kriteria kebaikan model peramalan yang dihasilkan nantinya, dan menyajikan output berupa grafik, angka, atau teks.

### C. Visualisasi Menggunakan Shiny

Dalam melakukan visualisasi hasil prediksi menggunakan *Shiny*, digunakan *syntax* pada Gambar 8. Gambar 8 menunjukkan langkah memunculkan hasil visualisasi dengan *Shiny* menggunakan UI dan Server yang sudah dibuat dalam tahap sebelumnya. Hasil visualisasi ditunjukkan oleh Gambar 9.

Visualisasi prediksi gempa pada Gambar 4 berisi, antara lain: (a) *Dropdown "Display"* untuk memilih mode tampilan data yang terdiri dari *Historical* dan *Forecast*. Mode *Historical* digunakan untuk menampilkan frekuensi gempa yang telah terjadi pada peta. Sedangkan mode *Forecast* digunakan untuk menampilkan hasil prediksi frekuensi gempa pada peta; (b) *Dropdown "Year"* untuk memilih tahun yang akan ditampilkan riwayat atau hasil prediksinya pada peta; (c) *Slider "Week"* untuk memilih minggu yang akan ditampilkan riwayat atau hasil prediksinya pada peta; (d) Grafik garis yang menampilkan data *Historical* (garis merah) dan data *Forecast* (garis hitam); (e) Nilai error dari hasil prediksi berdasarkan wilayah yang dipilih, terdiri dari ME (*Mean Error*), RMSE (*Root Mean Square Error*), dan MAE (*Mean Absolute Error*); (f) *Choropleth map* untuk menampilkan peta yang telah diarahkan ke wilayah Indonesia dan terdapat pembagian wilayah berdasarkan data gempa. Setiap wilayah memiliki intensitas warna yang berbeda-beda berdasarkan frekuensi gempa pada wilayah tersebut dan waktu yang dipilih. Keterangan hubungan intensitas warna dengan frekuensi gempa diletakkan pada bagian kanan bawah peta.

## V. KESIMPULAN

Hasil yang didapatkan adalah aplikasi ini mampu menunjukkan frekuensi gempa yang sudah terjadi di Indonesia dalam kurun waktu 5 tahun, yaitu mulai Februari 2016 hingga Februari 2021 dan mampu memprediksi frekuensi gempa yang akan di Indonesia hingga tahun 2022.

## DAFTAR PUSTAKA

- [1] Afnimar, *Seismologi*, 1st ed. Bandung: Institut Teknologi Bandung, 2009.