

**APLIKASI VISUALISASI TEOREMA LIMIT PUSAT**  
***CENTRAL LIMIT THEOREM (CLT) VISUALIZATION APP***

Disusun untuk memenuhi tugas mata kuliah Analisis Peubah Ganda

**Kelompok 5**

**Team 4KS1:**

Nida Annisa Miftahul Firdausi (11.6815)

Galang Retno Winarko (12.7151)

Joseph Christoffel (12.7202)

M. Arif Maulana (12.7231)

Mukhlis (12.7272)

Shabrina Izzati (12.7379)

Tiara Ratna Dewi (12.7403)

**Team 4KS2:**

Atikasari (12.7057)

Istiqomah Novi Purnanti (12.7194)

Muhammad Arief Eko Pratama (12.7261)

Nadya Amalia (12.7276)

Oemar Syarief Wibisono (12.7307)

Rivan Destyanugraha (12.7351)

Rizki Ramadhani Arif Trilana (12.7357)

Yohanes Eki Apriliawan (12.7438)

**SEKOLAH TINGGI ILMU STATISTIK**

**JAKARTA**

**2016**

## INTRODUCTION

Program ini dibuat untuk memenuhi tugas mata kuliah Analisis Peubah Ganda semester 8 Sekolah Tinggi Ilmu Statistik. Program ini merupakan visualisasi dari teorema limit pusat (*Central Limit Theorem*) yang menyebutkan bahwa vektor rata-rata dari setiap gugus sampel yang diambil dari sejumlah observasi yang bersifat independen akan selalu mendekati sebaran distribusi teoritis normal untuk jumlah sampel yang besar.

Pada aplikasi ini terdapat empat distribusi teoritis yang bisa dipilih yaitu distribusi normal, gamma, beta, dan uniform. Terdapat slider untuk mengubah ukuran sampel (*sample size*) dan jumlah gugus sampel (*number of samples*). Kemudian dari setiap gugus sampel dihitung rata-ratanya dan digambarkan dalam plot histogram pada panel sebelah kanan dari aplikasi.

Pada pembuatan program ini digunakan bahasa pemrograman R yang dilengkapi dengan package tcltk2 untuk pembuatan tampilan antarmuka (*Graphical User Interface/GUI*) dan package tkrplot dan ggplot2 untuk menampilkan grafik dalam aplikasi ini. Dikerjakan dengan bantuan aplikasi IDE RStudio, aplikasi ini dapat diselesaikan walaupun jauh dari kata sempurna. Kode dari aplikasi dan contoh tampilan dapat dilihat pada halaman berikutnya.

**Source code program visualisasi CLT pada R dengan GUI Tcl/Tk © Kelompok 5, 2016.**

```
library(tcltk2)
library(tkrplot)
library(ggplot2)

#normal
normal.dist <- function(){
  means <- rep(NA,nos)
  for(i in 1:nos){
    means[i] = mean(rnorm(n = ss,mean = par1,sd = par2))
  }
  meanss<-as.data.frame(means)
  dummy <- ggplot(data=meanss, aes(x=means)) +
  geom_histogram(aes(y=..density..,fill=..density..),

col="green",binwidth = 0.2) +
  geom_density() +
  ggtitle(paste("Population distribution: ",dis,"\n(miu=",par1,"",
sigma=",par2,")",
"\nSample size:",ss," Num of Samples:",nos)) +

stat_function(fun=dnorm,color="red",args=list(mean=mean(means),sd=sd(means
)))
  print(dummy)
}

#gamma
gamma.dist <- function() {
  xbar = rep(NA,nos)
  for(i in 1:nos){
    xbar[i] = mean(rgamma(ss, shape = par1, rate = par2))
  }
  xbars<-as.data.frame(xbar)
  dummy <- ggplot(data=xbars, aes(x=xbar)) +
  geom_histogram(aes(y=..density..,fill=..density..),

col="green",binwidth = 0.2) +
  geom_density() +
  ggtitle(paste("Population distribution: ",dis,"\n(shape=",par1,"",
rate=",par2,")",
"\nSample size:",ss," Num of Samples:",nos)) +

stat_function(fun=dnorm,color="red",args=list(mean=mean(xbar),sd=sd(xbar))
)
  print(dummy)
```

```
}

#Beta
beta.dist <- function() {
  data2 <- rep(0,nos)
  for(i in 1:nos) {
    data1 <- rbeta(n=ss,par1,par2)
    data2[i] <- mean(data1)
  }
  data2s <- as.data.frame(data2)
  dummy <- ggplot(data=data2s, aes(x=data2)) +
  geom_histogram(aes(y=..density..,fill=..density..),

  col="green",binwidth = 0.2) +
    geom_density() +
    ggtitle(paste("Population distribution: ",dis,"\n(alpha=",par1,"",
  beta=",par2,")",
              "\nSample size:",ss," Num of Samples:",nos)) +

  stat_function(fun=dnorm,color="red",args=list(mean=mean(data2),sd=sd(data2
  )))
  print(dummy)
}

#uniform
uniform.dist <- function() {
  xbar = rep(0,nos)
  for(i in 1:nos){
    xbar[i] = mean(runif(n = ss,min = par1,max = par2))
  }
  xbars<-as.data.frame(xbar)
  if(par1>par2){
    dummy <- ggtitle(paste("Lower bound=",par1,"is higher than upper
  bound=",par2,
                          ". Change your upper bound higher!"))
  } else {
    dummy <- ggplot(data=xbars, aes(x=xbar)) +
    geom_histogram(aes(y=..density..,fill=..density..),
                  col="green",
  binwidth = 0.2) +
      geom_density() + ggtitle(paste("Population distribution:
  ",dis,"\n(min=",par1,"", max=",par2,")",
                                  "\nSample size:",ss," Num of
  Samples:",nos)) +
```

```
stat_function(fun=dnorm,color="red",args=list(mean=mean(xbar),sd=sd(xbar))
)

}
print(dummy)
}

CLTvis <-function() {
  tclRequire("Bwidget")
  ss <- 30
  nos <- 200
  dis <- "Normal"
  par1 <- 0
  par2 <- 1

  winPlot <- tktoplevel()
  tktitle(winPlot) <- "CLT Visualization (c)Group 5"
  # Define a frame inside 'winPlot'
  winPlot$env$frm <- tk2frame(winPlot, borderwidth = 2, relief = "sunken",
                             padding = 0)

  img <- tkrplot::tkrplot(winPlot,fun=normal.dist,1.4,1.4)
  tkpack(img, side = "right",expand=FALSE, fill = "both")

  tkpack(winPlot$env$frm, expand = TRUE, fill = "both")

  winPlot$env$frm$frmDist <- tk2frame(winPlot$env$frm,padding=0)
  winPlot$env$frm$frmDistSelect <- tk2frame(winPlot$env$frm,padding=0)
  winPlot$env$frm$frmSamp <- tk2frame(winPlot$env$frm,padding=0)

  labelTextNormM <- tclVar(paste("mean"))
  sliderValueNormM <- tclVar("0")
  sliderValueLabelNormM <- tklabel(winPlot$env$frm$frmDist,
text=labelTextNormM)
  tkconfigure(sliderValueLabelNormM, textvariable=sliderValueNormM)
  sliderNormM <- tkcale(winPlot$env$frm$frmDist, from=-50, to=50,
showvalue=T,variable=sliderValueNormM,
                      resolution=1, orient="horizontal",length=400)
  sliderNormM.label <- tk2label(winPlot$env$frm$frmDist, text = "Mean:")

  labelTextNormSD <- tclVar(paste("standard deviation"))
  sliderValueNormSD <- tclVar("1")
  sliderValueLabelNormSD <- tklabel(winPlot$env$frm$frmDist,
text=labelTextNormSD)
  tkconfigure(sliderValueLabelNormSD, textvariable=sliderValueNormSD)
```

## Central Limit Theorem (CLT) Visualization

© 2016

```
sliderNormSD <- tkscale(winPlot$env$frm$frmDist, from=1, to=30,
showvalue=T,variable=SliderValueNormSD,
                        resolution=1, orient="horizontal",length=400)
sliderNormSD.label <- tk2label(winPlot$env$frm$frmDist, text = "Standard
Deviation:")

distribVal <- c("Normal", "Gamma", "Beta", "Uniform")
distribSel <- tclvar("Normal")
distribField <- ttkcombobox(winPlot$env$frm$frmDistSelect,
values=distribVal, textvariable=distribSel,
                        state="readonly")

onOK <- function(...) {
  dis <- tclvalue(distribSel)
  tkdestroy(winPlot$env$frm$frmDist)
  tkgrid.remove(winPlot$env$frm$frmSamp)
  if (dis == "Normal") {
    winPlot$env$frm$frmDist <- tk2frame(winPlot$env$frm,padding=0)

tkgrid(winPlot$env$frm$frmDist,sticky="we",columnspan=1,padx=0,pady=0,rows
pan=1,ipadx=0,ipady=0)

    par1 <- 0
    par2 <- 1
    img <- tkrreplot(img,fun=normal.dist,1.4,1.4)
  } else if(dis == "Gamma") {
    winPlot$env$frm$frmDist <- tk2frame(winPlot$env$frm,padding=0)

tkgrid(winPlot$env$frm$frmDist,sticky="we",columnspan=1,padx=0,pady=0,rows
pan=1,ipadx=0,ipady=0)

    par1 <- 1
    par2 <- 1
    img <- tkrreplot(img,fun=gamma.dist,1.4,1.4)
  } else if(dis == "Beta") {
    winPlot$env$frm$frmDist <- tk2frame(winPlot$env$frm,padding=0)

tkgrid(winPlot$env$frm$frmDist,sticky="we",columnspan=1,padx=0,pady=0,rows
pan=1,ipadx=0,ipady=0)

    par1 <- 1
    par2 <- 1

    img <- tkrreplot(img,fun=beta.dist,1.4,1.4)
  } else if(dis == "Uniform") {
    winPlot$env$frm$frmDist <- tk2frame(winPlot$env$frm,padding=0)
```

```
tkgrid(winPlot$env$frm$frmDist,sticky="we",columnspan=1,padx=0,pady=0,rows
pan=1,ipadx=0,ipady=0)
```

```
par1 <- 0
```

```
par2 <- 1
```

```
img <- tkrreplot(img,fun=uniform.dist,1.4,1.4)
} else {
  tkmessageBox(title="Missing...", message = "Select a distribution!")
}
```

```
tkgrid(winPlot$env$frm$frmSamp,sticky="we",columnspan=1,padx=0,pady=0,rows
pan=1,ipadx=0,ipady=0)
}
```

```
OKbutton <- tkbutton(winPlot$env$frm$frmDistSelect , text = "Set",
foreground = "darkgreen",
width = "12", command = onOK, default = "active",
borderwidth = 3)
lbl <- tk2label(winPlot$env$frm$frmDistSelect, text = "Population
distribution: ")
```

```
tkgrid(winPlot$env$frm$frmDistSelect,sticky="we",columnspan=3,padx=0,pady=
0,rowspan=1,ipadx=0,ipady=0)
tkgrid(lbl, distribField, OKbutton, padx = 10, pady = c(5,10))
```

```
tkgrid(winPlot$env$frm$frmDist,sticky="we",columnspan=1,padx=0,pady=0,rows
pan=1,ipadx=0,ipady=0)
```

```
tkgrid(sliderNormM.label, columnspan = 5, padx = 10, pady = c(5,2))
tkgrid(sliderNormM,columnspan = 5, padx = 10, pady = c(5,10))
```

```
tkgrid(sliderNormSD.label, columnspan = 5, padx = 10, pady = c(5,2))
tkgrid(sliderNormSD,columnspan = 5, padx = 10, pady = c(5,50))
```

```
tkgrid(winPlot$env$frm$frmSamp,sticky="we",columnspan=1,padx=0,pady=0,rows
pan=1,ipadx=0,ipady=0)
```

```
labelText3 <- tclVar(paste("sample size"))
sliderValue3 <- tclVar("30")
sliderValueLabel3 <- tklabel(winPlot$env$frm$frmSamp, text=labelText3)
tkconfigure(sliderValueLabel3, textvariable=sliderValue3)
```

```
slider3 <- tkscale(winPlot$env$frm$frmSamp, from=10, to=100,
showvalue=T,variable=SliderValue3,resolution=1,
orient="horizontal",length=400)
slider3.label <- tk2label(winPlot$env$frm$frmSamp, text = "Sample
size:")
tkgrid(slider3.label, columnspan = 5, padx = 10, pady = c(50,2))
tkgrid(slider3,columnspan = 5, padx = 10, pady = c(5,10))

labelText4 <- tclvar(paste("number of samples"))
sliderValue4 <- tclvar("200")
sliderValueLabel4 <- tklabel(winPlot$env$frm$frmSamp, text=labelText4)
tkconfigure(sliderValueLabel4, textvariable=SliderValue4)
slider4 <- tkscale(winPlot$env$frm$frmSamp, from=1, to=1000,
showvalue=T,variable=SliderValue4,resolution=1,
orient="horizontal",length=400)
slider4.label <- tk2label(winPlot$env$frm$frmSamp, text = "Number of
Samples:")
tkgrid(slider4.label, columnspan = 5, padx = 10, pady = c(5,2))
tkgrid(slider4,columnspan = 5, padx = 10, pady = c(5,10))

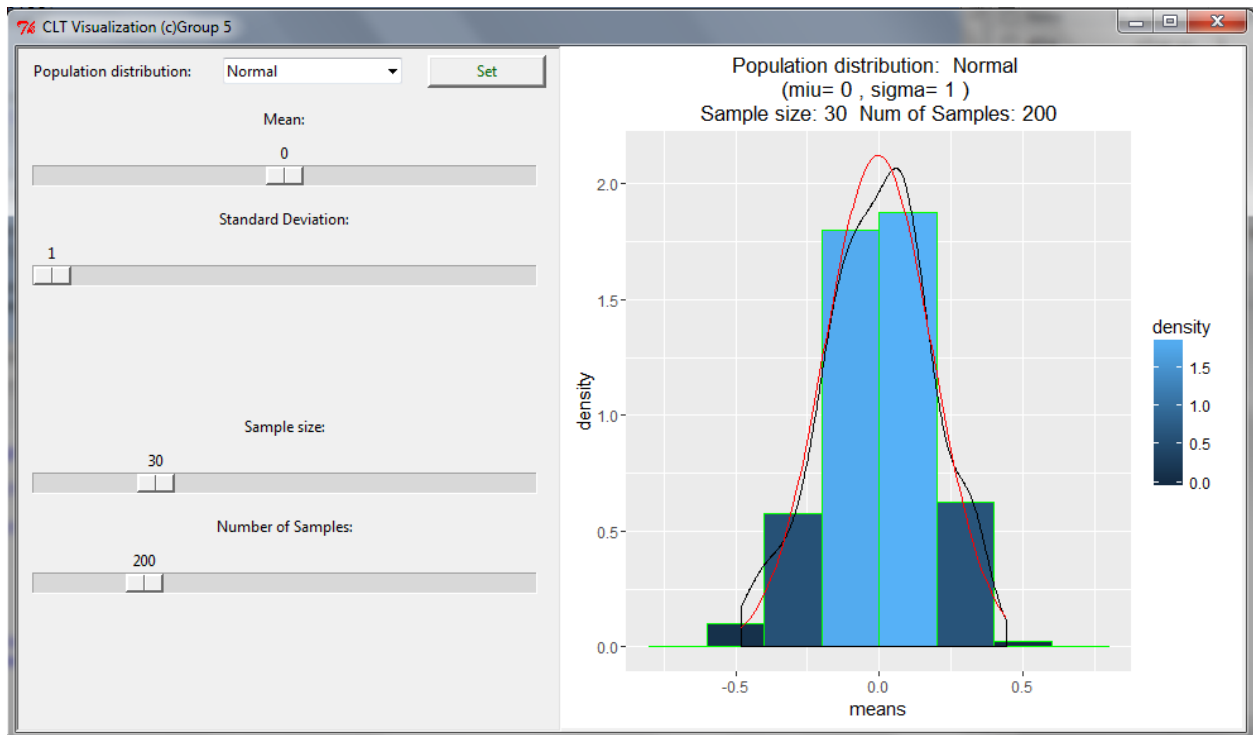
draw <- function(...){
  ss <- as.numeric(tclvalue(SliderValue3))
  nos <- as.numeric(tclvalue(SliderValue4))
  par1 <- as.numeric(tclvalue(SliderValueNormM))
  par2 <- as.numeric(tclvalue(SliderValueNormSD))
  img <- tkrreplot(img,fun=normal.dist,1.4,1.4)
}
tkconfigure(sliderNormM, command = draw)
tkconfigure(sliderNormSD, command = draw)
tkconfigure(slider3, command = draw)
tkconfigure(slider4, command = draw)
}
CLTvis()
```



### Contoh Output Program © Kelompok 5, 2016

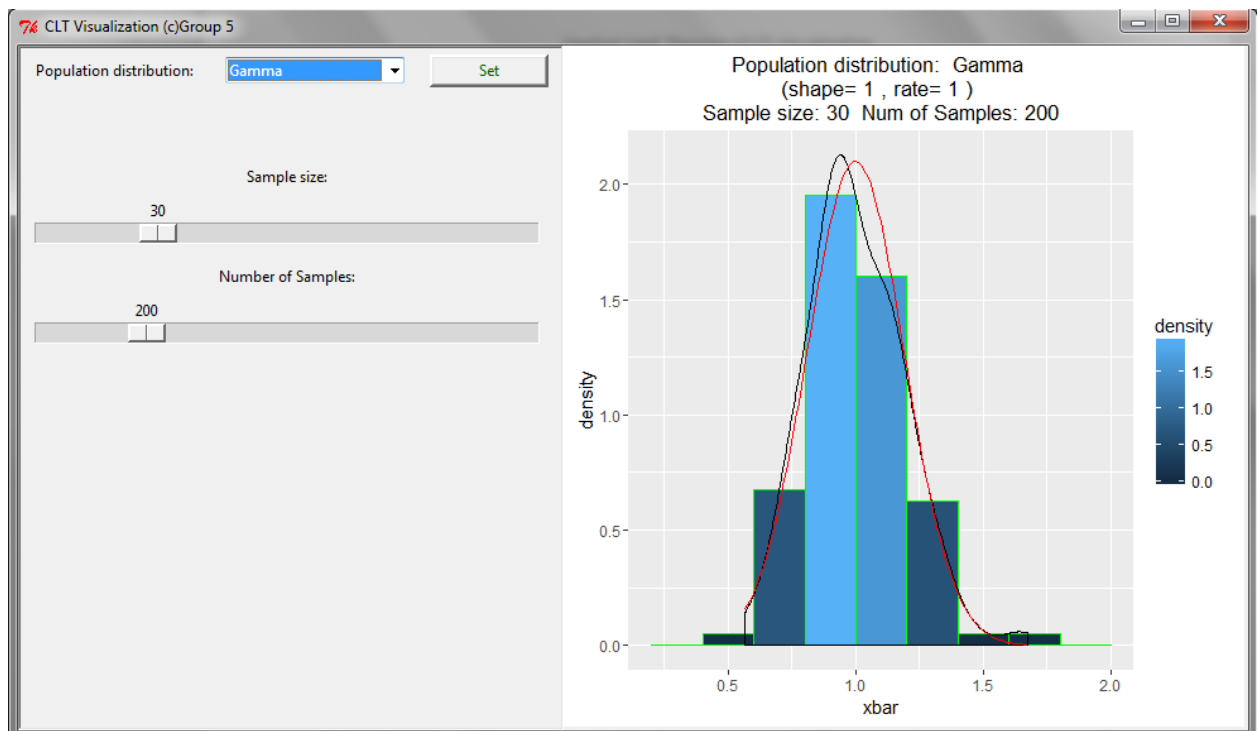
Langkah-langkah penggunaan program ini adalah:

1. Salin kode program pada program R dan *compile* hingga muncul tampilan sebagai berikut. Tampilan awal menunjukkan visualisasi untuk populasi berdistribusi Normal dengan rata-rata sebesar 0 dan standar deviasi sebesar 1.

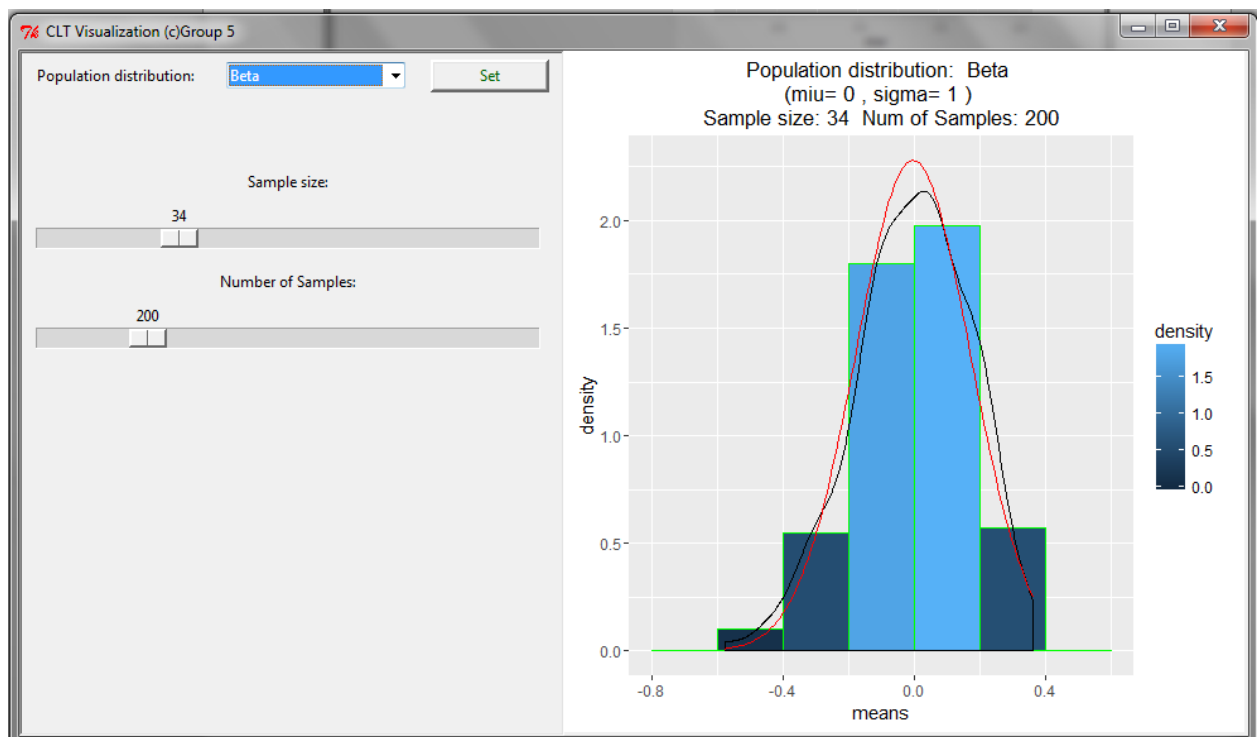


2. Geser *slider* untuk mengubah nilai parameter, ukuran sampel, maupun jumlah paket sampel. Setiap pergeseran akan secara otomatis mengubah tampilan plot. Perbedaan warna pada histogram menunjukkan density atau kepekatan data, semakin terang warna biru menunjukkan semakin banyak data yang berada pada kelompok tersebut. Garis warna hitam menunjukkan plot garis dari densitas data simulasi yang digunakan sedangkan garis warna merah menunjukkan plot garis dari distribusi teoritis normal.
3. Untuk memilih distribusi lain, pilih pada kotak pilihan dan kemudian tekan tombol Set. Pilihan distribusi yang disediakan berupa Normal, Gamma, Beta, dan Uniform. Contoh tampilan setiap distribusi dapat dilihat pada halaman berikut.

Central Limit Theorem (CLT) Visualization  
© 2016

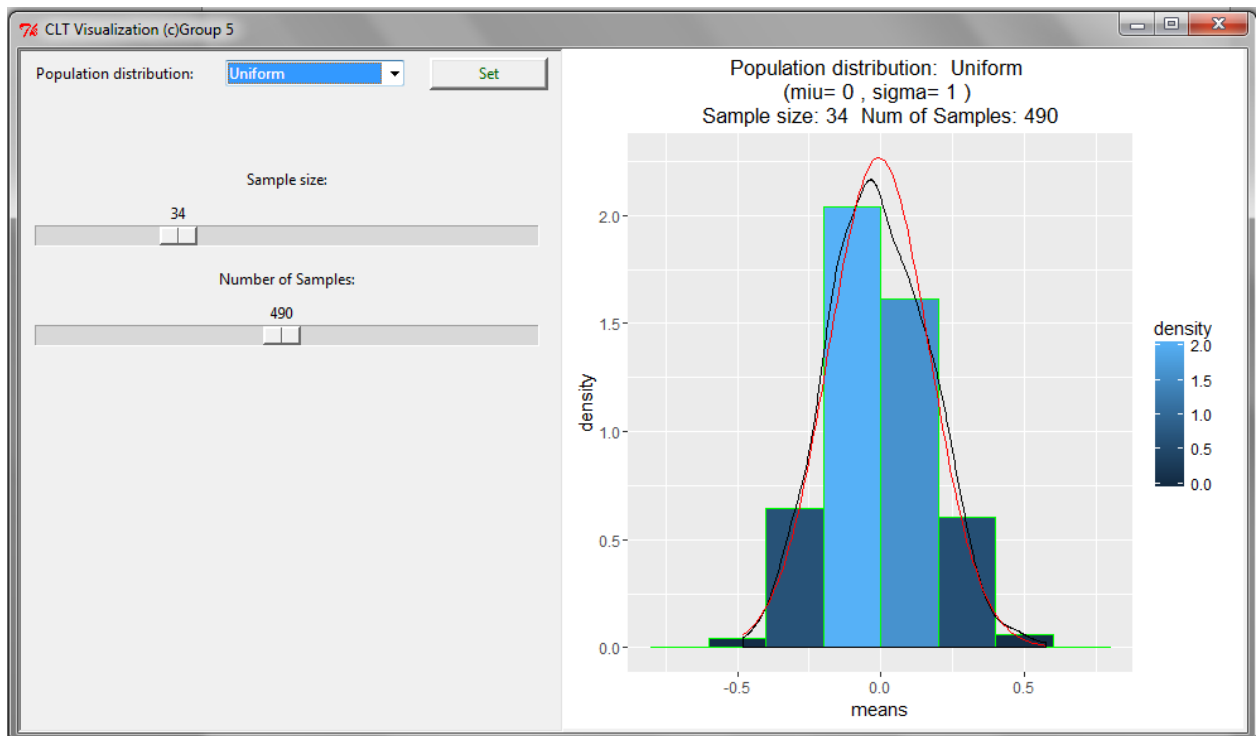


Tampilan visualisasi teori CLT untuk distribusi Gamma.



Tampilan untuk visualisasi CLT pada distribusi Beta

*Central Limit Theorem (CLT) Visualization*  
© 2016



Tampilan untuk visualisasi CLT pada distribusi Uniform.