

# Mini R Program

Irfan Fadil

30 Agustus 2020

## 1. Menghitung Jumlah Data

```
jumlah <- function(data){  
  sums = 0 # Mendefinisikan variabel Jumlah data  
  for (i in 1:length(data)){ # Melakukan penjumlah tiap data secara berulang  
    sums = sums+ data[i]  
  }  
  return(sums)  
}  
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan data  
jumlah(data) # Mencoba program
```

```
## [1] 62
```

## 2. Menghitung rata-rata

```
rata_rata <- function(data){  
  sums = 0 # Mendefinisikan variabel Jumlah data  
  for (i in 1:length(data)){  
    sums = sums + data[i]  
  }  
  # Menghitung Rataan: jumlah data dibagi banyak data  
  rataan=(sums/length(data))  
  return(rataan)  
}  
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan data  
rata_rata(data) # Mencoba program
```

```
## [1] 4.769231
```

## 3. Menghitung rata-rata (versi 2)

```
rata_rata <- function(data){  
  # Menghitung jumlah data menggunakan fungsi **jumlah(data)** pada program 1  
  sums= jumlah(data)  
  # Menghitung Rataan: jumlah data dibagi banyak data  
  rataan=(sums/length(data))  
  return(rataan)  
}
```

```
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan data
rata_rata(data) # Mencoba program
```

```
## [1] 4.769231
```

#### 4. Mencari Maksimum dari suatu data

```
maksimum <- function(data){
  # Mendefinisikan variabel nilai maksimum dari data
  # Nilai awal **maks** adalah data pertama
  maks = data[1]
  for (i in 2:length(data)){
    if (maks > data[i]){
      maks = maks
    }
    else {
      maks = data[i]
    }
  }
  return(maks)
}
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan data
maksimum(data) # Mencoba Program
```

```
## [1] 7
```

#### 5. Mencari Minimum dari suatu data

```
minimum <- function(data){
  # Mendefinisikan variabel nilai maksimum dari data: minn
  # Nilai awal **minn** adalah data pertama
  minn = data[1]
  for (i in 2:length(data)){
    if (minn < data[i]){
      minn = minn
    }
    else {
      minn = data[i]
    }
  }
  return(minn)
}
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan data
minimum(data) # Mencoba Program
```

```
## [1] 2
```

## 6. Mengurutkan Data

```
urutkan <- function(data, dari_yang = "terkecil"){
  data_urut = c() #Mendefinisikan variabel data yang diurutkan: data_urut
  #Mendefinisikan kode pengurutan data dari yang terkecil:dari_yang="terkecil"
  if (dari_yang == "terkecil"){
    for (j in 1:(length(data)-1)){
      #Mencari nilai minimum dan memasukkan ke dalam data_urut yang ke-i
      minn = data[1]
      for (i in 2:length(data)){
        if (minn < data[i]){
          minn = data[i]
        }
        else {
          minn = data[i]
        }
      }
      data_urut[j] = minn
    }
    for (k in 1:length(data)){
      if (data[k]==minn){
        data=data[-k]
        break
      }
      else {
        next
      }
    }
  }
  data_urut[length(data_urut)+1] = data[1]
}

#Mendefinisikan kode pengurutan data dari yang terbesar:dari_yang="terbesar"
else if (dari_yang=="terbesar") {
  for (j in 1:(length(data)-1)){
    #Mencari nilai minimum dan memasukkan ke dalam data_urut yang ke-i
    maks = data[1]
    for (i in 2:length(data)){
      if (maks > data[i]){
        maks = data[i]
      }
      else {
        maks = data[i]
      }
    }
    data_urut[j] =maks
  }
  for (k in 1:length(data)){
    if (data[k]==maks){
      data=data[-k]
      break
    }
    else {
      next
    }
  }
}
```

```

    }
    data_urut[length(data_urut)+1] = data[1]
  }
  return(data_urut)
}
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan data
# Mencoba Program mengurutkan data dari yang terkecil
urutkan(data, dari_yang = "terkecil")

## [1] 2 3 4 4 4 5 5 5 6 6 6 7

# Mencoba Program mengurutkan data dari yang terbesar
urutkan(data, dari_yang = "terbesar")

## [1] 7 6 6 6 5 5 5 5 4 4 4 3 2

```

## 7. Menghitung Modulo

```

modulo <- function(a,b){
  repeat {
    a=a-b #mengurangi penyebut
    if (a<b){
      break
    }
  }
  return(a)
}
modulo(1000,7) # Mencoba Program

## [1] 6

modulo(48,5) # Mencoba Program

## [1] 3

```

## 8. Menghitung Median

```

medians <- function(data){
  #mengurutkan data
  data_urut=c()
  for (j in 1:(length(data)-1)){
    minn = data[1]
    for (i in 2:length(data)){
      if (minn < data[i]){
        minn = data[i]
      }
      else {
        minn = minn
      }
    }
    data_urut[j] = minn
  }
  for (k in 1:length(data)){
    if (data[k]==minn){

```

```

        data=data[-k]
        break
    }
    else {
        next
    }
}
}
data_urut[length(data_urut)+1] = data[1]
a= length(data_urut) #Mendefinisikan panjang data ke variabel a
# Mencari Modulo panjang data yang dibagi dengan 2.
repeat {
    a=a-2
    if (a<2){
        break
    }
}
#Menghitung Median
if (a==1){
    #Menghitung median jika panjang kelas ganjil
    med = data_urut[(length(data_urut)/2)+1]
}
else {
    #Menghitung median jika panjang kelas genap
    med = (data_urut[(length(data_urut)/2)]+data_urut[(length(data_urut)/2)+1])/2
}
return(med)
}
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan Data
medians(data) # Mencoba program

## [1] 5

```

## 8. Menghitung Median (versi 2)

```

medians <- function(data){
    #mengurutkan data dengan fungsi urutkan pada program nomor 6
    data_urut=urutkan(data)
    # Mencari Modulo menggunakan fungsi modulo ada program nomor 7: a
    a = modulo(length(data),2)
    #Menghitung Median
    if (a==1){
        #Menghitung median jika panjang kelas ganjil
        med = data_urut[(length(data_urut)/2)+1]
    }
    else {
        #Menghitung median jika panjang kelas genap
        med = (data_urut[(length(data_urut)/2)]+data_urut[(length(data_urut)/2)+1])/2
    }
    return(med)
}
data <- c(2,4,5,6,4,5,6,7,6,5,3,4,5) # Mendefinisikan Data

```

```
medians(data) # Mencoba program
```

```
## [1] 5
```

## 9. Mencari Modus

```
modus <- function (data){  
  #mengurutkan data  
  data_urut = c()  
  for (j in 1:(length(data)-1)){  
    minn = data[1]  
    for (i in 2:length(data)){  
      if (minn < data[i]){  
        minn = data[i]  
      }  
      else {  
        minn = data[i]  
      }  
    }  
    data_urut[j] = minn  
  }  
  for (k in 1:length(data)){  
    if (data[k]==minn){  
      data=data[-k]  
      break  
    }  
    else {  
      next  
    }  
  }  
}   
data_urut[length(data_urut)+1] = data[1]  
#Menghitung frekuensi setiap data yang unik  
frekuensi=c(1)  
for (i in 2:length(data_urut)){  
  if (data_urut[i]==data_urut[i-1]){  
    frekuensi[i]=frekuensi[i-1]+1  
  }  
  else{  
    frekuensi[i]=1  
  }  
}  
maks = frekuensi[1]  
for (i in 2:length(frekuensi)){  
  if (maks > frekuensi[i]){  
    maks = maks  
  }  
  else {  
    maks = frekuensi[i]  
  }  
}  
return(data_urut[frekuensi==maks])  
}
```

```
data <- c(2,3,4,3,4,3,3,4,3,4,6,7,6,5,7,6) # Mendefinisikan data
modus(data) # Mencoba Program
```

```
## [1] 3
```

## 10. Mencari Modus (versi 2)

```
modus <- function (data){
  #mengurutkan data menggunakan fungsi **urutkan** pada program nomor 6
  data_urut = urutkan(data)
  #Menghitung frekuensi setiap data yang unik
  frekuensi=c(1)
  for (i in 2:length(data_urut)){
    if (data_urut[i]==data_urut[i-1]){
      frekuensi[i]=frekuensi[i-1]+1
    }
    else{
      frekuensi[i]=1
    }
  }
  maks = frekuensi[1]
  for (i in 2:length(frekuensi)){
    if (maks > frekuensi[i]){
      maks = maks
    }
    else {
      maks = frekuensi[i]
    }
  }
  return(data_urut[frekuensi==maks])
}
data <- c(2,3,4,3,4,3,3,4,3,4,6,7,6,5,7,6) # Mendefinisikan data
modus(data) # Mencoba Program
```

```
## [1] 3
```

## 11. Menghitung Variansi

```
variansi <- function(data){
  #menghitung Jumlah data (untuk menghitung rataaan)
  jum1 = 0
  for (i in 1:length(data)){
    jum1 = jum1 + data[i]
  }
  # Menghitung rataaan (jumlah_data/length(data))
  rataaan=(jum1/length(data))
  # Menghitung Variansi
  a = (data-rataaan)^2
  sums = 0
  for (i in 1:length(a)){
    sums = sums+ a[i]
  }
}
```

```

}
varian = sums/(length(data)-1)
return(varian)
}
data <- c(2,3,4,3,4,3,4,3,4,3,4,6,7,6,5,7,6) #mendefinisikan Data
variansi(data) #Mencoba Program

## [1] 2.367647

```

## 12. Menghitung Variansi (versi 2)

```

variansi <- function(data){
  # Menghitung rata-rata dengan fungsi rata_rata pada program 2.
  rata-rata=rata_rata(data)
  a = (data-rata-rata)^2
  #menghitung jumlah a :sums
  sums = jumlah(a)
  varian = sums/(length(data)-1)
  return(varian)
}
data <- c(2,3,4,3,4,3,4,3,4,3,4,6,7,6,5,7,6) #mendefinisikan Data
variansi(data) #Mencoba Program

## [1] 2.367647

```

## 13. Menghitung Standar Deviasi

```

standar_deviasi <- function(data){
  #menghitung Jumlah data (untuk menghitung rata-rata)
  juml = 0
  for (i in 1:length(data)){
    juml = juml + data[i]
  }
  # Menghitung rata-rata (jumlah_data/length(data))
  rata-rata=(juml/length(data))
  a = (data-rata-rata)^2
  sums = 0
  for (i in 1:length(a)){
    sums = sums+ a[i]
  }
  # Menghitung Standar Deviasi
  stdev = sqrt(sums/(length(data)-1))
  return(stdev)
}
data <- c(2,3,4,3,4,3,4,3,4,3,4,6,7,6,5,7,6) #mendefinisikan Data
standar_deviasi(data) #Mencoba Program

## [1] 1.538716

#Menghitung standar deviasi juga bisa dengan sqrt(variansi(data))
sqrt(variansi(data))

```



## [1] 1.538716