

Generate data berdistribusi Weibull dengan miu 0.5 dan beta 1.5

```

n1<-100
n2<-200
n3<-400
beta<-1.5
miu<-0.5
ulangan<-1000

X100<-matrix(0,n1,ulangan)
for (k in 1:ulangan)
{
    X100[,k]<-rweibull(n1,beta,miu)
}

data.X100=data.frame(X100)
write.table(data.X100,file="D:/R/random1.50.5_X100.txt",sep= ,col.names = NA)

X200<-matrix(0,n2,ulangan)
for (k in 1:ulangan)
{
    X200[,k]<-rweibull(n2,beta,miu)
}

data.X200=data.frame(X200)
write.table(data.X200,file="D:/R/random1.50.5_X200.txt",sep= ,col.names = NA)

X400<-matrix(0,n3,ulangan)
for (k in 1:ulangan)
{
    X400[,k]<-rweibull(n3,beta,miu)
}

data.X400=data.frame(X400)
write.table(data.X400,file="D:/R/random1.50.5_X400.txt",sep= ,col.names = NA)

```

Generate data berdistribusi Weibull dengan miu 1.5 dan beta 1

```

n1<-100
n2<-200
n3<-400
beta<-1.5
miu<-1
ulangan<-1000

```

```

X100<-matrix(0,n1,ulangan)
for (k in 1:ulangan)
{
    X100[,k]<-rweibull(n1,beta,miu)
}

data.X100=data.frame(X100)
write.table(data.X100,file="D:/R/random1.51_X100.txt",sep= ,col.names = NA)

X200<-matrix(0,n2,ulangan)
for (k in 1:ulangan)
{
    X200[,k]<-rweibull(n2,beta,miu)
}

data.X200=data.frame(X200)
write.table(data.X200,file="D:/R/random1.51_X200.txt",sep= ,col.names = NA)

X400<-matrix(0,n3,ulangan)
for (k in 1:ulangan)
{
    X400[,k]<-rweibull(n3,beta,miu)
}

data.X400=data.frame(X400)
write.table(data.X400,file="D:/R/random1.51_X400.txt",sep= ,col.names = NA)

```

Generate data berdistribusi Weibull dengan miu 0.5 dan beta 1

```

n1<-100
n2<-200
n3<-400
beta<-1
miu<-0.5
ulangan<-1000

X100<-matrix(0,n1,ulangan)
for (k in 1:ulangan)
{
    X100[,k]<-rweibull(n1,beta,miu)
}

data.X100=data.frame(X100)
write.table(data.X100,file="D:/R/random10.5_X100.txt",sep= ,col.names = NA)

```

```

X200<-matrix(0,n2,ulangan)
for (k in 1:ulangan)
{
    X200[,k]<-rweibull(n2,beta,miu)
}

data.X200=data.frame(X200)
write.table(data.X200,file="D:/R/random10.5_X200.txt",sep= ,col.names = NA)

X400<-matrix(0,n3,ulangan)
for (k in 1:ulangan)
{
    X400[,k]<-rweibull(n3,beta,miu)
}

data.X400=data.frame(X400)
write.table(data.X400,file="D:/R/random10.5_X400.txt",sep= ,col.names = NA)

```

Generate data berdistribusi Weibull dengan miu 1 dan beta 1

```

n1<-100
n2<-200
n3<-400
beta<-1
miu<-1
ulangan<-1000

X100<-matrix(0,n1,ulangan)
for (k in 1:ulangan)
{
    X100[,k]<-rweibull(n1,beta,miu)
}

data.X100=data.frame(X100)
write.table(data.X100,file="D:/R/random11_X100.txt",sep= ,col.names = NA)

X200<-matrix(0,n2,ulangan)
for (k in 1:ulangan)
{
    X200[,k]<-rweibull(n2,beta,miu)
}

data.X200=data.frame(X200)
write.table(data.X200,file="D:/R/random11_X200.txt",sep= ,col.names = NA)

```

```

X400<-matrix(0,n3,ulangan)
for (k in 1:ulangan)
{
    X400[,k]<-rweibull(n3,beta,miu)
}

data.X400=data.frame(X400)
write.table(data.X400,file="D:/R/random11_X400.txt",sep= ,col.names = NA)

```

salah satu coding untung menduga nilai parameter (μ, β) dengan $n = 100$ dengan perulangan sebanyak 1000 yang berdistribusi Weibull pada *full data*

- **Menduga parameter dengan $miu = 0.5$ dan $beta = 1.5$**

```

beta<-1.5
miu<-0.5
ulangan<-1000
iterasi<-100
X100<-read.table(file = "D:/R/random1.50.5_X100.txt",header=TRUE)

# melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program
ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100
x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya
pengulangan

n1<-100
A<-array(0,c(n1,iterasi,ulangan))
B<-array(0,c(n1,1,ulangan))
C<-array(0,c(n1, iterasi,ulangan))
D<-array(0,c(n1, iterasi,ulangan))
Sum_A<-array(0,c(1, iterasi,ulangan))
Sum_C<-array(0,c(1, iterasi,ulangan))
Sum_D<-array(0,c(1, iterasi,ulangan))

b_0<-matrix(0,1,iterasi)
b_d100<-matrix(c(beta,b_0),1,101)
b_duga100<-array(b_d100,c(1,101,ulangan))

for (j in 1:ulangan)
{
for (k in 1:iterasi)
{
for (i in 1:n1)
{

```

```

A[i,k,j]<-X100[i,j]^b_duga100[1,k,j]
B[i,1,j]<-log(X100[i,j])
C[i,k,j]<-A[i,k,j]*B[i,1,j]
D[i,k,j]<-A[i,k,j]*(B[i,1,j]^2)
}
Sum_A[1,k,j]<-sum(A[,k,j])
Sum_C[1,k,j]<-sum(C[,k,j])
Sum_D[1,k,j]<-sum(D[,k,j])
b_duga100[1,k+1,j]<-b_duga100[1,k,j]-(((Sum_C[1,k,j]/Sum_A[1,k,j])-
(1/b_duga100[1,k,j])-(sum(B[,1,j])/n1))/(((Sum_A[1,k,j]*Sum_D[1,k,j])-
(Sum_C[1,k,j]^2))/(Sum_A[1,k,j]^2)))+(1/b_duga100[1,k,j]^2)))
}
}

X100_sb<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
X100_sb[1,j] <- sum(X100[,j]^b_duga100[1,100,j])
}

miu_duga100<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
miu_duga100[1,j] <- (X100_sb[1,j]/n1)^(1/b_duga100[1,100,j])
}

data.Beta_dugaX100=data.frame(b_duga100[1,100,])
write.table(data.Beta_dugaX100,file="D:/R/Bduga_X100.txt",sep= ,col.names = NA)

data.Miu_dugaX100=data.frame(miu_duga100)
write.table(data.Miu_dugaX100,file="D:/R/MIUduga_X100.txt",sep= ,col.names =
NA)

```

Coding penduga parameter (μ, β) yang berdistribusi Weibull untuk data tersensor kanan dengan $n = 100$, intensitas tersensor (1%, 5%, 10%, 15%), dengan perulangan sebanyak 1000

Karena program yang akan dilampirkan akan sangat banyak, maka penulis hanya melampirkan program dengan intensitas tersensor sebesar 1%, dan untuk kasus intensitas tersensor 5%, 10%, dan 15%, penulis hanya menuliskan untuk kasus $\mu = 1$ dan $\beta = 1$ dengan $n = 100$

- **Menduga parameter dengan $\mu = 0.5$, $\beta = 1.5$, dengan intensitas tersensor 1%**

```

X100 <- read.table (file = "D:/R/random1.50.5_X100.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=100

#tersensor kanan 1%

beta<-1.5
miu<-0.5
n1<-100
r <-99
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X100_sort <-matrix(0,n1,ulangan)
for (j in 1:ulangan)
{
    X100_sort[,j] <- sort(X100[,j])
}

Xi<- matrix(0,n1-1,ulangan)
for (j in 1:ulangan)
{
    for (i in 1:99)
    {
        Xi[i,j] <- X100_sort[i,j]
    }
}

Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
    Xr[1,j]<-Xi[99,j]
}

# melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program
ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100

```

x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

beta_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}

for (j in 1:ulangan)
{
  q[1,j] <- sum(log(Xi[,j]))
}

for (j in 1:ulangan)
{
  s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
  t[1,j] <- (n1-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
  beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  y[1,j] <- (n1-r) * ((Xr[1,j])^beta_duga[1,j])
}

```

```

for (j in 1:ulangan)
{
  z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
  miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_1.50.5_X100.txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_1.50.5_X100.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $miu = 1$, $beta = 1.5$, dengan intensitas tersensor 1%**

```

X100 <- read.table (file = "D:/R/random1.51_X100.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=100

#tersensor kanan 1 %

beta<-1.5
miu<-1
n1<-100
r <-99
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X100_sort <-matrix(0,n1,ulangan)
for (j in 1:ulangan)
{
  X100_sort[,j] <- sort(X100[,j])
}

Xi<- matrix(0,n1-1,ulangan)
for (j in 1:ulangan)

```



```

{
  for (i in 1:99)
  {
    Xi[i,j] <- X100_sort[i,j]
  }
}
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
  Xr[1,j]<-Xi[99,j]
}

```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

```

```

beta_duga<-matrix(0,1,ulangan)

```

```

for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}

```

```

for (j in 1:ulangan)
{
  q[1,j] <- sum(log(Xi[,j]))
}

```

```

for (j in 1:ulangan)
{
  s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

```

```

for (j in 1:ulangan)
{
  t[1,j] <- (n1-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

```

```

for (j in 1:ulangan)

```

```

{
  beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  y[1,j] <- (n1-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
  z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
  miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_1.51_X100.
txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_1.51_X100
.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $miu = 0.5$, $beta = 1$, dengan intensitas tersensor 1%**

```
X100 <- read.table (file = "D:/R/random10.5_X100.txt",header=TRUE)
```

```
#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=100
```

```
#tersensor kanan 1 %
```

```
beta<-1
miu<-0.5
n1<-100
r <-99
```

```
ulangan <- 1000
```

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung pada intensitas tersensor yang telah di tentukan sebelumnya

```
X100_sort <-matrix(0,n1,ulangan)
for (j in 1:ulangan)
{
  X100_sort[,j] <- sort(X100[,j])
}
```

```
Xi<- matrix(0,n1-1,ulangan)
for (j in 1:ulangan)
{
  for (i in 1:99)
  {
    Xi[i,j] <- X100_sort[i,j]
  }
}
```

```
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
  Xr[1,j]<-Xi[99,j]
}
```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```
p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)
```

```
beta_duga<-matrix(0,1,ulangan)
```

```
for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}
```

```
for (j in 1:ulangan)
{
```

```

        q[1,j] <- sum(log(Xi[,j]))
    }
for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}
for (j in 1:ulangan)
{
    t[1,j] <- (n1-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}
for (j in 1:ulangan)
{
    beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n1-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
    z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
    miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_10.5_X100.
txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_10.5_X100
.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 1$, $\beta = 1$, dengan intensitas tersensor 1%**

```

X100 <- read.table (file = "D:/R/random11_X100.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=100

#tersensor kanan 1 %

beta<-1
miu<-1
n1<-100
r <-99
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X100_sort <-matrix(0,n1,ulangan)
for (j in 1:ulangan)
{
    X100_sort[,j] <- sort(X100[,j])
}

Xi<- matrix(0,n1-1,ulangan)
for (j in 1:ulangan)
{
    for (i in 1:99)
    {
        Xi[i,j] <- X100_sort[i,j]
    }
}

Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
    Xr[1,j]<-Xi[99,j]
}

# melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program
ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100
x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya
pengulangan

```

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

beta_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    p[1,j] <- r * log(miu)
}

for (j in 1:ulangan)
{
    q[1,j] <- sum(log(Xi[,j]))
}

for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
    t[1,j] <- (n1-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
    beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n1-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{

```

```

        z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
    }
    for (j in 1:ulangan)
    {
        miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
    }

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_11_X100.txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_11_X100.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $miu = 0.5$, $beta = 1.5$, dengan intensitas tersensor 1%**

```

X200 <- read.table (file = "D:/R/random1.50.5_X200.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=200

#tersensor kanan 1 %

beta<-1.5
miu<-0.5
n2<-200
r <-198
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X200_sort <-matrix(0,n2,ulangan)
for (j in 1:ulangan)
{
    X200_sort[,j] <- sort(X200[,j])
}

Xi<- matrix(0,n2-2,ulangan)
for (j in 1:ulangan)
{
    for (i in 1:198)

```

```

        {
            Xi[i,j] <- X200_sort[i,j]
        }
    }
    Xr<-matrix(0,1,ulangan)
    for (j in 1:ulangan)
    {
        Xr[1,j]<-Xi[198,j]
    }

```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

```

```

beta_duga<-matrix(0,1,ulangan)

```

```

for (j in 1:ulangan)
{
    p[1,j] <- r * log(miu)
}

```

```

for (j in 1:ulangan)
{
    q[1,j] <- sum(log(Xi[,j]))
}

```

```

for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

```

```

for (j in 1:ulangan)
{
    t[1,j] <- (n2-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

```

```

for (j in 1:ulangan)
{

```



```

        beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
    }

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n2-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
    z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
    miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_1.50.5_X200.txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_1.50.5_X200.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 1$, $\beta = 1.5$, dengan intensitas tersensor 1%**

```
X200 <- read.table (file = "D:/R/random1.51_X200.txt",header=TRUE)
```

```
#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=200
```

```
#tersensor kanan 1 %
```

```
beta<-1.5
```

```
miu<-1
```

```
n2<-200
```

```
r <-198
```

```
ulangan <- 1000
```

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung pada intensitas tersensor yang telah di tentukan sebelumnya

```
X200_sort <-matrix(0,n2,ulangan)
for (j in 1:ulangan)
{
  X200_sort[,j] <- sort(X200[,j])
}
```

```
Xi<- matrix(0,n2-2,ulangan)
for (j in 1:ulangan)
{
  for (i in 1:198)
  {
    Xi[i,j] <- X200_sort[i,j]
  }
}
```

```
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
  Xr[1,j]<-Xi[198,j]
}
```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```
p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)
```

```
beta_duga<-matrix(0,1,ulangan)
```

```
for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}
```

```
for (j in 1:ulangan)
{
  q[1,j] <- sum(log(Xi[,j]))
}
```

```

}

for (j in 1:ulangan)
{
  s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
  t[1,j] <- (n2-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
  beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  y[1,j] <- (n2-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
  z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
  miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_1.51_X200.
txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_1.51_X200
.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 0.5$, $\beta = 1$, dengan intensitas tersensor 1%**

```

X200 <- read.table (file = "D:/R/random10.5_X200.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=200

#tersensor kanan 1 %

beta<-1
miu<-0.5
n2<-200
r <-198
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X200_sort <-matrix(0,n2,ulangan)
for (j in 1:ulangan)
{
    X200_sort[,j] <- sort(X200[,j])
}

Xi<- matrix(0,n2-2,ulangan)
for (j in 1:ulangan)
{
    for (i in 1:198)
    {
        Xi[i,j] <- X200_sort[i,j]
    }
}

Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
    Xr[1,j]<-Xi[198,j]
}

# melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program
ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100
x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya
pengulangan

```

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

beta_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    p[1,j] <- r * log(miu)
}

for (j in 1:ulangan)
{
    q[1,j] <- sum(log(Xi[,j]))
}

for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
    t[1,j] <- (n2-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
    beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n2-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{

```

```

        z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
    }
    for (j in 1:ulangan)
    {
        miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
    }

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_10.5_X200.
txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_10.5_X200
.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 1$, $\beta = 1$, dengan intensitas tersensor 1%**

```

X200 <- read.table (file = "D:/R/random11_X200.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=200

#tersensor kanan 1 %

beta<-1
miu<-1
n2<-200
r <-198
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X200_sort <-matrix(0,n2,ulangan)
for (j in 1:ulangan)
{
    X200_sort[,j] <- sort(X200[,j])
}

Xi<- matrix(0,n2-2,ulangan)
for (j in 1:ulangan)
{
    for (i in 1:198)
    {

```

```

        Xi[i,j] <- X200_sort[i,j]
    }
}
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
    Xr[1,j]<-Xi[198,j]
}

```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

```

```

beta_duga<-matrix(0,1,ulangan)

```

```

for (j in 1:ulangan)
{
    p[1,j] <- r * log(miu)
}

```

```

for (j in 1:ulangan)
{
    q[1,j] <- sum(log(Xi[,j]))
}

```

```

for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

```

```

for (j in 1:ulangan)
{
    t[1,j] <- (n2-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

```

```

for (j in 1:ulangan)
{

```

```

        beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
    }

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n2-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
    z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
    miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_11_X200.txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_11_X200.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 0.5$, $\beta = 1.5$, dengan intensitas tersensor 1%**

```
X400 <- read.table (file = "D:/R/random1.50.5_X400.txt",header=TRUE)
```

```
#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=400
```

```
#tersensor kanan 1 %
```

```
beta<-1.5
```

```
miu<-0.5
```

```
n3<-400
```

```
r <-396
```



```
ulangan <- 1000
```

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung pada intensitas tersensor yang telah di tentukan sebelumnya

```
X400_sort <-matrix(0,n3,ulangan)
for (j in 1:ulangan)
{
  X400_sort[,j] <- sort(X400[,j])
}
```

```
Xi<- matrix(0,n3-4,ulangan)
for (j in 1:ulangan)
{
  for (i in 1:396)
  {
    Xi[i,j] <- X400_sort[i,j]
  }
}
```

```
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
  Xr[1,j]<-Xi[396,j]
}
```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```
p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)
```

```
beta_duga<-matrix(0,1,ulangan)
```

```
for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}
```

```

for (j in 1:ulangan)
{
  q[1,j] <- sum(log(Xi[,j]))
}

for (j in 1:ulangan)
{
  s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
  t[1,j] <- (n3-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
  beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  y[1,j] <- (n3-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
  z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
  miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_1.50.5_X40
0.txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)

```

```
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_1.50.5_X400.txt",sep=,col.names = NA)
```

- **Menduga parameter dengan $\mu = 1$, $\beta = 1.5$, dengan intensitas tersensor 1%**

```
X400 <- read.table (file = "D:/R/random1.51_X400.txt",header=TRUE)
```

```
#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=400
```

```
#tersensor kanan 1 %
```

```
beta<-1.5
```

```
miu<-1
```

```
n3<-400
```

```
r <-396
```

```
ulangan <- 1000
```

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung pada intensitas tersensor yang telah di tentukan sebelumnya

```
X400_sort <-matrix(0,n3,ulangan)
```

```
for (j in 1:ulangan)
```

```
{
    X400_sort[,j] <- sort(X400[,j])
}
```

```
Xi<- matrix(0,n3-4,ulangan)
```

```
for (j in 1:ulangan)
```

```
{
    for (i in 1:396)
    {
        Xi[i,j] <- X400_sort[i,j]
    }
}
```

```
Xr<-matrix(0,1,ulangan)
```

```
for (j in 1:ulangan)
```

```
{
    Xr[1,j]<-Xi[396,j]
}
```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

beta_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}

for (j in 1:ulangan)
{
  q[1,j] <- sum(log(Xi[,j]))
}

for (j in 1:ulangan)
{
  s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
  t[1,j] <- (n3-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
  beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)

```

```

{
  y[1,j] <- (n3-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
  z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
  miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_1.51_X400.
txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_1.51_X400
.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 0.5$, $\beta = 1$, dengan intensitas tersensor 1%**

```

X400 <- read.table (file = "D:/R/random10.5_X400.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=400

#tersensor kanan 1 %

beta<-1
miu<-0.5
n3<-400
r <-396
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X400_sort <-matrix(0,n3,ulangan)
for (j in 1:ulangan)
{
  X400_sort[,j] <- sort(X400[,j])
}

```

```

Xi<- matrix(0,n3-4,ulangan)
for (j in 1:ulangan)
{
  for (i in 1:396)
  {
    Xi[i,j] <- X400_sort[i,j]
  }
}
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
  Xr[1,j]<-Xi[396,j]
}

```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

```

```

beta_duga<-matrix(0,1,ulangan)

```

```

for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}

```

```

for (j in 1:ulangan)
{
  q[1,j] <- sum(log(Xi[,j]))
}

```

```

for (j in 1:ulangan)
{
  s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

```

```

for (j in 1:ulangan)

```

```

{
  t[1,j] <- (n3-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
  beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  y[1,j] <- (n3-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
  z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
  miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_10.5_X400.
txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_10.5_X400
.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $miu = 1$, $beta = 1$, dengan intensitas tersensor 1%**

```
X400 <- read.table (file = "D:/R/random11_X400.txt",header=TRUE)
```

```
#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=400
```

```
#tersensor kanan 1 %
```

```

beta<-1
miu<-1
n3<-400
r <-396
ulangan <- 1000

```

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung pada intensitas tersensor yang telah di tentukan sebelumnya

```

X400_sort <-matrix(0,n3,ulangan)
for (j in 1:ulangan)
{
    X400_sort[,j] <- sort(X400[,j])
}

```

```

Xi<- matrix(0,n3-4,ulangan)
for (j in 1:ulangan)
{
    for (i in 1:396)
    {
        Xi[i,j] <- X400_sort[i,j]
    }
}

```

```

Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
    Xr[1,j]<-Xi[396,j]
}

```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

```

```

beta_duga<-matrix(0,1,ulangan)

```



```

for (j in 1:ulangan)
{
    p[1,j] <- r * log(miu)
}

for (j in 1:ulangan)
{
    q[1,j] <- sum(log(Xi[,j]))
}

for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
    t[1,j] <- (n3-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
    beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n3-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
    z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
    miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

```

```
data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_1%_11_X400.txt",sep=,col.names = NA)
```

```
data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_1%_11_X400.txt",sep=,col.names = NA)
```

Coding penduga parameter (μ, β) yang berdistribusi Weibull untuk data tersensor kanan dengan $n = 100$, intensitas tersensor 5%, dengan perulangan sebanyak 1000

- **Menduga parameter dengan $miu = 1$, $beta = 1$, dengan intensitas tersensor 5%**

```
X100 <- read.table (file = "D:/R/random11_X100.txt",header=TRUE)
```

```
#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=100
```

```
#tersensor kanan 5 %
```

```
beta<-1
miu<-1
n1<-100
r <-95
ulangan <- 1000
```

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung pada intensitas tersensor yang telah di tentukan sebelumnya

```
X100_sort <-matrix(0,n1,ulangan)
for (j in 1:ulangan)
{
  X100_sort[,j] <- sort(X100[,j])
}
```

```
Xi<- matrix(0,n1-5,ulangan)
for (j in 1:ulangan)
{
  for (i in 1:95)
  {
```

```

        Xi[i,j] <- X100_sort[i,j]
    }
}
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
    Xr[1,j]<-Xi[95,j]
}

```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

```

```
beta_duga<-matrix(0,1,ulangan)
```

```

for (j in 1:ulangan)
{
    p[1,j] <- r * log(miu)
}

```

```

for (j in 1:ulangan)
{
    q[1,j] <- sum(log(Xi[,j]))
}

```

```

for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

```

```

for (j in 1:ulangan)
{
    t[1,j] <- (n1-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

```

```

for (j in 1:ulangan)
{

```

```

        beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
    }

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n1-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
    z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
    miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

#rata-rata penduga parameter

mean_betaduga <- matrix(0,1,1)

{
    mean_betaduga <- sum(beta_duga[1,])/ulangan
}

mean_miuduga <- matrix(0,1,1)

{
    mean_miuduga <- sum(miu_duga[1,])/ulangan
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_5%_11_X100.txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_5%_11_X100.txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 1$, $\beta = 1$, dengan intensitas tersensor 10%**

```

X100 <- read.table (file = "D:/R/random11_X100.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=100

#tersensor kanan 10 %

beta<-1
miu<-1
n1<-100
r <-90
ulangan <- 1000

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu
menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung
pada intensitas tersensor yang telah di tentukan sebelumnya

X100_sort <-matrix(0,n1,ulangan)
for (j in 1:ulangan)
{
    X100_sort[,j] <- sort(X100[,j])
}

Xi<- matrix(0,n1-10,ulangan)
for (j in 1:ulangan)
{
    for (i in 1:90)
    {
        Xi[i,j] <- X100_sort[i,j]
    }
}

Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
    Xr[1,j]<-Xi[90,j]
}

# melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program
ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100
x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya
pengulangan

```

```

p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)

beta_duga<-matrix(0,1,iterasi)

for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}

for (j in 1:ulangan)
{
  q[1,j] <- sum(log(Xi[,j]))
}

for (j in 1:ulangan)
{
  s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}

for (j in 1:ulangan)
{
  t[1,j] <- (n1-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}

for (j in 1:ulangan)
{
  beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
  y[1,j] <- (n1-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)

```

```

{
  z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
  miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

#rata-rata penduga parameter

mean_betaduga <- matrix(0,1,1)

{
  mean_betaduga <- sum(beta_duga[1,])/ulangan
}

mean_miuduga <- matrix(0,1,1)

{
  mean_miuduga <- sum(miu_duga[1,j])/ulangan
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_10%_11_X100.t
xt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_10%_11_X100.
txt",sep= ,col.names = NA)

```

- **Menduga parameter dengan $\mu = 1$, $\beta = 1$, dengan intensitas tersensor 10%**

```

X100 <- read.table (file = "D:/R/random11_X100.txt",header=TRUE)

#pendugaan parameter (beta,lamda) data tersensor kanan dengan n=100

#tersensor kanan 15 %

beta<-1
miu<-1
n1<-100
r <-85
ulangan <- 1000

```

#proses penyensoran data, dimana data harus di urutkan terlebih dahulu menggunakan fungsi sort, kemudian tentukan banyaknya data tersensor bergantung pada intensitas tersensor yang telah di tentukan sebelumnya

```
X100_sort <-matrix(0,n1,ulangan)
for (j in 1:ulangan)
{
  X100_sort[,j] <- sort(X100[,j])
}
```

```
Xi<- matrix(0,n1-15,ulangan)
for (j in 1:ulangan)
{
  for (i in 1:85)
  {
    Xi[i,j] <- X100_sort[i,j]
  }
}
```

```
Xr<-matrix(0,1,ulangan)
for (j in 1:ulangan)
{
  Xr[1,j]<-Xi[85,j]
}
```

melakukan pendugaan parameter (beta,miu) dengan n=100 dimana pada program ini menggunakan fungsi array, dimana matriks yang akan terbuat adalah matriks 100 x 1000 dengan seratus adalah banyaknya data dan 1000 adalah banyaknya pengulangan

```
p<-matrix(0,1,ulangan)
q<-matrix(0,1,ulangan)
s<-matrix(0,1,ulangan)
t<-matrix(0,1,ulangan)
```

```
beta_duga<-matrix(0,1,ulangan)
```

```
for (j in 1:ulangan)
{
  p[1,j] <- r * log(miu)
}
```

```
for (j in 1:ulangan)
{
```



```

        q[1,j] <- sum(log(Xi[,j]))
    }
for (j in 1:ulangan)
{
    s[1,j] <- sum( ((Xi[,j]/miu)^beta) * (log(Xi[,j]/miu)) )
}
for (j in 1:ulangan)
{
    t[1,j] <- (n1-r) * ((Xr[1,j]/miu)^beta) * (log(Xr[1,j]/miu))
}
for (j in 1:ulangan)
{
    beta_duga[1,j] <- r / (p[1,j] - q[1,j] + s[1,j] - t[1,j])
}

#penduga miu

y<-matrix(0,1,ulangan)
z<-matrix(0,1,ulangan)
miu_duga<-matrix(0,1,ulangan)

for (j in 1:ulangan)
{
    y[1,j] <- (n1-r) * ((Xr[1,j])^beta_duga[1,j])
}

for (j in 1:ulangan)
{
    z[1,j] <- sum((Xi[,j])^beta_duga[1,j])
}
for (j in 1:ulangan)
{
    miu_duga[1,j] <- ((1/r)*(y[1,j] + z[1,j]))^1/beta_duga[1,j]
}

#rata-rata penduga parameter

mean_betaduga <- matrix(0,1,1)

{
    mean_betaduga <- sum(beta_duga[1,])/ulangan
}

```

```

mean_miuduga <- matrix(0,1,1)

{
mean_miuduga <- sum(miu_duga[1,])/ulangan
}

data.Miu_dugasensor1_X100=data.frame(miu_duga)
write.table(data.Miu_dugasensor1_X100,file="D:/R/miudugasensor_15%_11_X100.txt",sep= ,col.names = NA)

data.Beta_dugasensor1_X100=data.frame(beta_duga)
write.table(data.Beta_dugasensor1_X100,file="D:/R/betadugasensor_15%_11_X100.txt",sep= ,col.names = NA)

```

- **Lampiran Coding untuk plot distribusi Weibull pada nilai $\mu_1=1$, $\mu_2=0.5$, dan $\beta=1$**

```

miu1<-1
miu2<-0.5

beta<-1
m<-1000
n<-1000
x<-seq(-pi, pi, len = 65)
x<-array(0,c(n,1))
x1<-array(0,c(n,1))
x2<-array(0,c(n,1))

fw1<-array(0,c(n,1))
fw2<-array(0,c(n,1))

for(i in 1:n)
{
for(j in 1:x1)
{
x[i]<-i/10
fw1[i]<-beta*(miu1^(-beta))*(x[i]^(beta-1))*(exp(-((x[i]/miu1)^beta)))
}
}
for(i in 1:n)
{
for(j in 1:x2)
{
x[i]<-i/10

```

```
fw2[i]<-beta*(miu2^(-beta))*(x[i]^(beta-1))*(exp(-((x[i]/miu2)^beta)))
}
}
```

```
plot(x,fw1,type="l",xlim=range(0,20),ylim=range(0,2.5),xlab="time",ylab="Pdf
Weibull",col="green",lty=1)
lines(x, fw2,col="red", lty=2)
```

```
temp<-legend("topright",legend=c("beta=1 miu1=1","beta=1 miu2=0.5"),col =
c("green","red","blue"),text.col = "green4", lty = c(1, -1, 0), pch = c(-1, 3, 21),merge
= TRUE, bg = 'gray90')
```

- **Lampiran Coding untuk plot distribusi Weibull pada nilai miu1=1, miu2=0.5, dan beta=1.5**

```
miu1<-1
miu2<-0.5
```

```
beta<-1.5
m<-1000
n<-1000
x<-seq(-pi, pi, len = 65)
x<-array(0,c(n,1))
x1<-array(0,c(n,1))
x2<-array(0,c(n,1))
```

```
fw1<-array(0,c(n,1))
fw2<-array(0,c(n,1))
```

```
for(i in 1:n)
{
for(j in 1:x1)
{
x[i]<-i/10
fw1[i]<-beta*(miu1^(-beta))*(x[i]^(beta-1))*(exp(-((x[i]/miu1)^beta)))
}
}
for(i in 1:n)
{
for(j in 1:x2)
{
x[i]<-i/10
fw2[i]<-beta*(miu2^(-beta))*(x[i]^(beta-1))*(exp(-((x[i]/miu2)^beta)))
}
}
}
```

```
plot(x,fw1,type="l",xlim=range(0,20),ylim=range(0,2.5),xlab="time",ylab="laju
kegagalan",col="green",lty=1)
lines(x, fw2,col="red", lty=2)
```

```
temp<-legend("topright",legend=c("beta=1.5 miu1=1","beta=1.5 miu2=0.5"),col =
c("green","red","blue"),text.col = "green4", lty = c(1, -1, 0), pch = c(-1, 3, 21),merge
= TRUE, bg = 'gray90')
```

- **Coding untuk plot distribusi Weibull pada nilai miu=0.5 beta1=1 dan beta2=1.5**

```
beta1<-1
beta2<-1.5
```

```
miu<-0.5
m<-1000
n<-1000
x<-seq(-pi, pi, len = 65)
x<-array(0,c(n,1))
x1<-array(0,c(n,1))
x2<-array(0,c(n,1))
```

```
fw1<-array(0,c(n,1))
fw2<-array(0,c(n,1))
```

```
for(i in 1:n)
{
for(j in 1:x1)
{
x[i]<-i/10
fw1[i]<-beta1*(miu^(-beta1))*(x[i]^(beta1-1))*(exp(-((x[i]/miu)^beta1)))
}
}
for(i in 1:n)
{
for(j in 1:x2)
{
x[i]<-i/10
fw2[i]<-beta2*(miu^(-beta2))*(x[i]^(beta2-1))*(exp(-((x[i]/miu)^beta2)))
}
}
```

```
plot(x,fw1,type="l",xlim=range(0,10),ylim=range(0,2),xlab="time",ylab="Pdf
Weibull",col="green",lty=1)
```

```
lines(x, fw2,col="red", lty=2)
```

```
temp<-legend("topright",legend=c("beta1=1 miu=0.5","beta2=1.5 miu=0.5"),col =
c("green","red","blue"),text.col = "black", lty = c(1, -1, 2), pch = c(-1, 3, 4),merge =
TRUE, bg = 'gray90')
```

- **Coding untuk plot distribusi Weibull pada nilai miu=1 beta1=1 dan beta2=1.5**

```
beta1<-1
beta2<-1.5
```

```
miu<-1
m<-1000
n<-1000
x<-seq(-pi, pi, len = 65)
x<-array(0,c(n,1))
x1<-array(0,c(n,1))
x2<-array(0,c(n,1))
```

```
fw1<-array(0,c(n,1))
fw2<-array(0,c(n,1))
```

```
for(i in 1:n)
{
for(j in 1:x1)
{
x[i]<-i/10
fw1[i]<-beta1*(miu^(-beta1))*(x[i]^(beta1-1))*(exp(-((x[i]/miu)^beta1)))
}
}
for(i in 1:n)
{
for(j in 1:x2)
{
x[i]<-i/10
fw2[i]<-beta2*(miu^(-beta2))*(x[i]^(beta2-1))*(exp(-((x[i]/miu)^beta2)))
}
}
}
```

```
plot(x,fw1,type="l",xlim=range(0,10),ylim=range(0,1),xlab="time",ylab="Pdf
Weibull",col="green",lty=1)
lines(x, fw2,col="red", lty=2)
```

```
temp<-legend("topright",legend=c("beta1=1 miu=1","beta2=1.5 miu=1"),col =  
c("green","red","blue"),text.col = "black", lty = c(1, -1, 2), pch = c(-1, 3, 4),merge =  
TRUE, bg = 'gray90')
```