

# **LAPORAN PRAKTIKUM KE – 4**

## **“ Ujian Tengah Praktikum “**

**Oleh :**

**A.Rofiqi Maulana (125090500111025)**



**Asisten :**

- 1. Angga Wahyu Pratama (115090500111061)**
- 2. Apriliantono (115090507111005)**

**LABORATORIUM STATISTIKA  
PROGRAM STUDI STATISTIKA  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2014**



## BAB IV

### HASIL DAN PEMBAHASAN

- 4.1 Bangkitkan data secara acak mengikuti sebaran binomial  $n = 50$ ,  $x = 10$ ,  $p = 0.5$ . Buatlah fungsi untuk menghitung simpangan baku, rata - rata, modus, dan kuartil

No	Source Code dan penjelasan
1	<code>x=rbinom(50,10,0.5)</code>
	Membuat bilangan acak yang mengikuti sebaran binomial dengan 50 observasi, banyaknya percobaan 10 kali dengan peluang sukses = 0.5
2	<code>#mencari simpangan baku</code>
	Memberikan komentar untuk menghitung simpangan baku
3	<code>std=function(x){</code>
	Membuat fungsi menghitung simpangan baku dengan nama fungsi std dari data x
4	<code>n=length(x);</code>
	Menyimpan banyaknya data ke variabel n
5	<code>ragam=(sum(x*x)-((sum(x)^2)/n))/(n-1)}</code>
	Menghitung ragam dari data x yaitu dengan rumus $Ragam = (\sum x^2 - (\sum x)^2/n) / (n - 1)$
6	<code>std=sqrt(ragam)</code>
	Menghitung simpangan baku yaitu akar dari ragam
7	<code>simpangan_baku=std(x)</code>
	Menyimpan nilai dari std(x) ke simpangan_baku
8	<code>simpangan_baku</code>
	Memanggil atau menampilkan hasil dari perhitungan simpangan baku
9	<code>#Mencari rata-rata</code>
	Membuat komentar dalam perhitungan rata-rata
10	<code>rata=function(x){</code>
	Membuat fungsi menghitung rata-rata dengan nama fungsi rata dari data x
11	<code>n=length(x);</code>
	Menyimpan banyaknya data ke variabel n
12	<code>rata=sum(x)/n}</code>
	Membuat perhitungan rata – rata yaitu $Rata-rata = \sum x/n$
13	<code>rata_rata=rata(x)</code>

	Menyimpan hasil perhitungan rata-rata ke variabel rata_rata
14	rata_rata
	Memanggil hasil dari perhitungan rata-rata dari data tersebut
15	#Mencari kuartil bawah
	Membuat komentar dalam perhitungan kuartil bawah (Q1)
16	q1=function(x){
	Membuat fungsi menghitung kuartil 1 dari suatu data x
17	n=length(x);odd=(n+1)%% 4;
	Menyimpan banyaknya data ke variabel n. odd sisa hasil bagi dari (n+1) dibagi dengan 4.
18	y=sort(x)
	Membuat variabel y yang digunakan untuk menyimpan data x yang telah diurutkan
19	if(odd==0){
	Merupakan kondisi saat sisa hasil bagi = 0 (artinya n+1 habis dibagi 4)
20	q1=(y[((n+1)/4)])}
	Mencari kuartil 1 dengan cara mencari data ke (n+1)/4 dari data yang telah diurutkan
21	else
	Merupakan kondisi saat odd tidak habis dibagi 4
22	q1=(y[floor((n+1)/4)]+y[ceiling((n+1)/4)]/2}
	Floor berfungsi untuk melakukan pembulatan ke bawah (misal 4.5 dibulatkan ke 4). Ceiling berfungsi melakukan pembulatan bilangan ke atas (misal 4.5 dibulatkan ke 5). Mencari kuartil 1 dengan cara $q1 = \text{Data ke } (n+1)/4$ . Jika $n = 6$ maka $q1 = \text{data ke } (6+1)/4 = \text{data ke } 1,75 = (\text{data ke } 1 + \text{data ke } 2)/2$ .
23	kuartil_bawah=q1(x)
	Menyimpan hasil perhitungan q1 ke variabel kuartil_bawah
24	kuartil_bawah Memanggil hasil perhitungan Q1 dari data tersebut
25	#mencari median
	Memberikan komentar dalam perhitungan median atau kuartil 2
26	med=function(x){
	Membuat fungsi menghitung kuartil 2 dengan nama fungsi med dari data x
27	n=length(x);odd=(n+1)%% 2;

	Menyimpan banyaknya data ke variabel n. odd sisa hasil bagi dari (n+1) dibagi dengan 2.
28	y=sort(x)
	Membuat variabel y yang digunakan untuk menyimpan data x yang telah diurutkan
29	if(odd==0){
	Merupakan kondisi saat sisa hasil bagi = 0 (artinya n+1 habis dibagi 2)
30	med=(y[(n+1)/2])}
	Menghitung nilai median yaitu data ke (n+1)/2
31	else
	Merupakan kondisi saat odd tidak habis dibagi 4
32	med=(y[floor(n+1)/2]+y[ceiling(n+1)/2])/2}
	Floor berfungsi untuk melakukan pembulatan ke bawah (misal 4.5 dibulatkan ke 4). Ceiling berfungsi melakukan pembulatan bilangan ke atas (misal 4.5 dibulatkan ke 5). Mencari median dengan cara med=Data ke (n+1)/2. Jika n = 6 maka med = data ke(6+1)/2 = data ke 3,5= (data ke-3 + data ke-5)/2.
33	#Mencari q3
	Memberikan komentar dalam perhitungan kuartil 3
34	n=length(x);odd=(n+1)% 4;
	Menyimpan banyaknya data ke variabel n. odd sisa hasil bagi dari (n+1) dibagi dengan 4.
35	y=sort(x)
	Membuat variabel y yang digunakan untuk menyimpan data x yang telah diurutkan
36	if(odd==0){
	Merupakan kondisi saat sisa hasil bagi = 0 (artinya n+1 habis dibagi 4)
37	q3=(y[3*(n+1)/4])}
	Mencari kuartil 3 dengan cara mencari data ke 3(n+1)/4 dari data yang telah diurutkan
38	else
	Merupakan kondisi saat odd tidak habis dibagi 4
39	q3=(y[floor(3*(n+1)/4)+y[ceiling(3*(n+1)/4)])/2}
	Floor berfungsi untuk melakukan pembulatan ke bawah (misal 4.5 dibulatkan ke 4). Ceiling berfungsi melakukan pembulatan bilangan ke atas (misal 4.5 dibulatkan ke 5). Mencari kuartil 3 dengan cara q3=Data ke 3(n+1)/4. Jika

	$n = 6$ maka $q1 = \text{data ke } 3(6+1)/4 = \text{data ke } 5,25 = (\text{data ke } 5 + \text{data ke } 6)/2$ .
40	kuartil_atas=q3(x)
	Menyimpan hasil perhitungan q3 ke variabel kuartil_atas
41	kuartil_atas
	Memanggil hasil perhitungan Q3 dari data tersebut
42	#mencari modus
	Memberikan komentar dalam perhitungan modus
43	$n=\text{length}(x)$
	Membuat variabel n yang digunakan untuk menyimpan banyaknya data (ukuran data)
44	for(i in 1:n){
	Melakukan perulangan dari i =1 sampai n
45	mo=0
	Proses inisiasi bahwa modus = 0
56	j=i+1
	Membuat variabel baru yaitu j dengan j= i+1 yang variabel j tersebut akan dilakuka perulangan
57	for(j in i:n){
	Melakukan perulangan dari j dimulai dari i sampai n
58	if(x[i]==x[j])
	Merupakan kondisi saat data ke i sama dengan data ke j
59	mo=mo+1 }
	Saat data ke i sama dengan data ke j, maka modus yaitu banyaknya data yang sama akan bertambah 1,2 ,3 dst sesuai dengan banyaknya data yang sama
60	mi[i]=mo }
	mi merupakan variabel yang menyimpan banyaknya bilangan yang sama (m0).mi[1] = 2 artinya data ke 2 mempunyai frekuensi yang sama sebanyak dua kali
61	mod=max(mi)
	Dari semua data, akan diperbandingkan frekuensi dan frekuensi terbesar yang muncul disimpan dalam variabel mod
62	for(k in 1:n){
	Melakukan perulangan k dari 1 sampai n.
63	if(mi[k]==mod)
	Merupaka kondisi saat frekuensi data ke k sama dengan mod.
64	modus=x[k]

	Data tersebut di cari yang menghasilkan frekuensi data ke k sama dengan mod. Dan modulus adalah data ke k yang mempunyai frekuensi paling besar
65	modus
	Memanggil hasil pencarian modulus data tersebut

Berikut merupakan perbandingan hasil perhitungan secara komputasi dan secara langsung menggunakan perintah di R.

#### Perintah langsung di R

Simpangan baku, rata-rata	<pre>&gt; mean(x) [1] 5.16 &gt; sd(x) [1] 1.718566</pre>
Kuartil	<pre>&gt; summary(x)   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   1.00   4.00   6.00   5.16   6.00   8.00 &gt; .</pre>
Modus	<pre>&gt; table(as.vector(x))   1  2  3  4  5  6  7  8  1  2  7  9  4 16  8  3 &gt;  </pre>

#### Perhitungan secara komputasi

Simpangan baku	<pre>&gt; x=rbinom(50,10,0.5) &gt; #mencari simpangan baku &gt; std=function(x){ +   n=length(x); +   ragam=(sum(x*x)-((sum(x)^2)/n))/(n-1) +   std=sqrt(ragam) + } &gt; simpangan_baku=std(x) &gt; simpangan_baku [1] 1.718566 &gt;</pre>
----------------	--

Rata – rata	<pre> &gt; #Mencari rata-rata &gt; rata=function(x){ +   n=length(x); +   rata=sum(x)/n + } &gt; rata_rata=rata(x) &gt; rata_rata [1] 5.16 &gt;   </pre>
Kuartil 1	<pre> &gt; #Mencari kuartil bawah &gt; q1=function(x){ +   n=length(x); odd=(n+1)%% 4; +   y=sort(x) +   if(odd==0){ +     q1=(y[((n+1)/4)]) +   } +   else +     q1=(y[floor((n+1)/4)]+y[ceiling((n+1)/4)])/2 + } &gt; kuartil_bawah=q1(x) &gt; kuartil_bawah [1] 4 &gt;   </pre>
Kuartil 2	<pre> &gt; #mencari median &gt; med=function(x){ +   n=length(x); odd=(n+1)%% 2; +   y=sort(x) +   if(odd==0){ +     med=(y[(n+1)/2]) +   } +   else +     med=(y[floor(n+1)/2]+y[ceiling(n+1)/2])/2 + } &gt; median=med(x) &gt; median [1] 6 &gt;   </pre>
Kuartil 3	<pre> &gt; q3=function(x){ +   n=length(x); odd=(n+1)%% 4; +   y=sort(x) +   if(odd==0){ +     q3=(y[3*(n+1)/4]) +   } +   else +     q3=(y[floor(3*(n+1)/4)]+y[ceiling(3*(n+1)/4)])/2 + } &gt; kuartil_atas=q3(x) &gt; kuartil_atas [1] 6 </pre>



Modus	<pre> &gt; #mencari modus &gt; n=length(x) &gt; for(i in 1:n){ +   mo=0 +   j=i+1 +   for(j in i:n){ +     if(x[i]==x[j]) +       mo=mo+1 +   } +   mi[i]=mo} &gt; mod=max(mi) &gt; for(k in 1:n){ +   if(mi[k]==mod) +     modus=x[k] + } &gt; modus [1] 6 &gt;   </pre>
-------	---

Didapatkan hasil perhtungan sebagai berikut

Rata-rata	5,16
Simpangan baku	1,718566
Q1	4
Q2	6
Q3	6
Modus	6

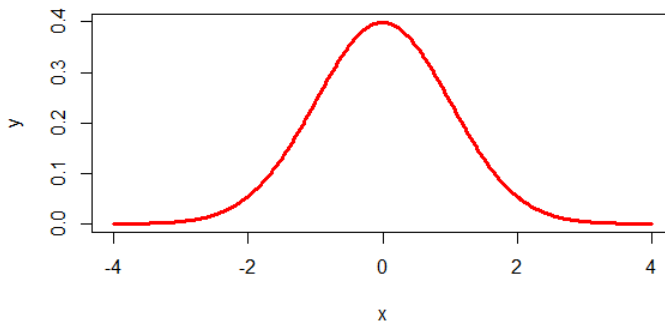
Dari perhitungan secara komputasi dan menggunakan perintah secara langsung di R , hasil yang didapatkan sama.

- 4.2      Buat grafik fungsi sebaran normal baku  
 Sebaran normal baku mempunyai rentang x dari  $-4$  sampai  $+4$ .

No	Source code dan penjelasan
1	<code>x=seq(-4,4,length=500)</code>
	Membuat rentang x dari -4 sampai 4 dengan banyak observasi = 500 (ukuran data = 500)
2	<code>y=dnorm(x,0,1)</code>

	Membuat fungsi kepekatan peluang dari sebaran normal baku ( rata –rata =0 dan ragam=1 )
3	<code>plot(x,y,type="l",lwd=3,col="red")</code>
	Membuat plot sebaran normal baku dengan dengan tipe garis line, ketebalan garis 3 titik, dan menggunakan warna merah

Berikut merupakan grafik sebaran normal baku



4.3 Hitung peluang sebaran normal baku untuk  $z=1.645$ ,  $z=1.96$ ,  $z=2.25$

No	Source Code dan penjelasan
1	<code>pnorm(q = 1.645,mean=0,sd=1,lower.tail=F)</code> Menghitung peluang sebaran normal baku $P(z>1.645)$ . <code>lower.tail=F</code> artinya peluang yang digunakan bukan komulatif, tapi peluang lebih dari 1.645
2	<code>pnorm(q = 1.96,mean=0,sd=1,lower.tail=F)</code> Menghitung peluang sebaran normal baku $P(z>1.96)$ . <code>lower.tail=F</code> artinya peluang yang digunakan bukan komulatif, tapi peluang lebih dari 1.96
3	<code>pnorm(q = 2.25,mean=0,sd=1,lower.tail=F)</code> Menghitung peluang sebaran normal baku $P(z>2.25)$ . <code>lower.tail=F</code> artinya peluang yang digunakan bukan komulatif, tapi peluang lebih dari 2.25

Berikut merupakan hasil perhitungan sebaran normal baku untuk beberapa nilai z tersebut

```
> pnorm(q = 1.645,mean=0,sd=1,lower.tail=F)
[1] 0.04998491
> pnorm(q = 1.96,mean=0,sd=1,lower.tail=F)
[1] 0.0249979
> pnorm(q = 2.25,mean=0,sd=1,lower.tail=F)
[1] 0.01222447
```

Dapat diringkas dalam suatu tabel

Nilai z	P(Z>z)
1.645	0.0499
1.96	0.024
2.25	0.0124

4.4 Hitung nilai z untuk peluang berikut  
p =0.1, p=0.01, p=0.05

No	Source code dan penjelasan
1	qnorm(0.1,mean=0,sd=1,lower.tail=F) Menghitung nilai nilai z sebaran normal baku dengan peluang 0.1. Menghitung z sedemikian hingga $P(Z>z) = 0.1$ . Lower.tail=F artinya peluang yang digunakan bukan komulatif, tapi peluang lebih dari $[P(Z>z)]$
2	qnorm(0.01,mean=0,sd=1,lower.tail=F) Menghitung nilai nilai z sebaran normal baku dengan peluang 0.1. Menghitung z sedemikian hingga $P(Z>z) = 0.01$ . Lower.tail=F artinya peluang yang digunakan bukan komulatif, tapi peluang lebih dari $[P(Z>z)]$
3	qnorm(0.05,mean=0,sd=1,lower.tail=F) Menghitung nilai nilai z sebaran normal baku dengan peluang 0.1. Menghitung z sedemikian hingga $P(Z>z) = 0.05$ . Lower.tail=F artinya peluang yang digunakan bukan komulatif, tapi peluang lebih dari $[P(Z>z)]$

Berikut merupakan hasil perhitungan nilai z dari beberapa peluang tersebut

```
> qnorm(0.1,mean=0,sd=1,lower.tail=F)
[1] 1.281552
> qnorm(0.01,mean=0,sd=1,lower.tail=F)
[1] 2.326348
>
> qnorm(0.05,mean=0,sd=1,lower.tail=F)
[1] 1.644854
> |
```

Dapat diringkas dalam suatu tabel

$P(Z > z)$	Nilai z
0.1	1.28
0.01	2.32
0.05	1.645

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

1. Perhitungan menggunakan perintah langsung di R dengan secara komputasi menghasilkan hasil akhir yang sama
2. Dalam membuat grafik sebaran normal baku, sumbu x adalah berkisar dari -4 sampai 4.
3. Secara umum, menghitung nilai peluang dari peluang sebaran normal baku adalah  
`pnorm(q,mean,sd=1,lower.tail=F)`  
dengan q adalah quantile
4. Secara umum, menghitung nilai z dari peluang sebaran normal baku adalah  
`qnorm(p,mean=0,sd=1,lower.tail=F)`  
dengan p adalah peluang sebaran normal baku

#### **5.2 Saran**

Dalam membuat plot suatu sebaran tertentu harus diperhatikan batas interval sumbu x. Karena akan menentukan densitas dari sebaran tersebut. Jika selang interval sumbu x tidak tepat, maka plot akan terlihat berbeda dari yang sebenarnya. Hati = hati juga dalam membuat plot menggunakan tipe garis lin atau histogram karena akan memberikan gambar yang berbeda.