
**JURNAL PRAKTIKUM
(LAB. ACTIVITY)
STATISTIKA
S1 TEKNIK INFORMATIKA**

Materi 5:
Ukuran Penyebaran

Oleh:

NAMA : Rizqy Mubarak

NIM : 23.0504.0024

Dosen:
Emilya Ullly Artha, M. Kom

**S1 - TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MAGELANG
2024**

Distribusi Frekuensi

A. Tujuan

Setelah praktikum ini, praktikan diharapkan dapat:

1. Memahami istilah distribusi frekuensi dan pengelompokan data dalam statistika
2. Dapat membuat tabel distribusi frekuensi
3. Dapat membuat visualisasi dari tabel distribusi frekuensi

B. Peralatan Praktikum

1. PC ataupun laptop yang telah terinstall program R dan RStudio
2. Telah selesai dalam pemahaman materi pada Lab Activity sebelumnya (ukuran sentral)
3. Memiliki pemahaman mengenai deret angka dan perhitungan nilai sentral

C. Praktikum

1. Mendefinisikan Data :

Menyimpan data dalam vector **data**.

```
> data <- c(75,80,30,70,20,35,65,65,70,57,  
+          55,25,58,70,40,35,36,45,40,25,  
+          15,55,35,65,40,15,30,30,45,40)  
> data  
[1] 75 80 30 70 20 35 65 65 70 57 55 25 58 70 40 35 36 45 40 25 15 55 35 65 40 15 30 30 45 40
```

2. Mengurutkan Data:

Menggunakan fungsi sort untuk mengurutkan data secara menaik.

```
> sort(data)  
[1] 15 15 20 25 25 30 30 30 35 35 35 36 40 40 40 40 45 45 55 55 57 58 65 65 65 70 70 70 75 80  
> |
```

3. Menentukan Jumlah Data (n)

Menggunakan length untuk menentukan jumlah data.

```
> n = length(data)  
> n  
[1] 30  
> |
```

4. Menentukan sebaran data, yaitu selisih antara nilai maksimum dan minimum dalam dataset.

Menggunakan fungsi range.

```
> range(data)  
[1] 15 80  
> #hitung jangkauan  
> diff(range(data))  
[1] 65  
> |
```

5. Menghitung Jumlah Kelas (K):

Menggunakan rumus Sturges, $K = 1 + 3.3 * \log_{10}(n)$, untuk menghitung jumlah kelas yang optimal.

```
> jumlahKelas = 1 + 3.3 * log10 (n)  
> jumlahKelas  
[1] 5.8745  
> |
```

6. Membulatkan Jumlah Kelas:

Melakukan pembulatan pada nilai K agar menjadi bilangan bulat.

```
> #lakukan pembulatan
> K = round(jumlahKelas)
> K
[1] 6
> |
```

7. Menghitung Interval Kelas:

Menggunakan rumus $\text{interval} = (\text{xmax} - \text{xmin}) / K$ untuk mendapatkan ukuran interval.

```
> #hitung interval
> interval = (xmax-xmin)/K
> interval
[1] 10.66667
> |
```

8. Membulatkan Interval:

Membulatkan nilai interval ke atas menggunakan `ceiling(interval)` untuk memastikan semua data tercover.

```
> p = ceiling(interval)
> p
[1] 11
> |
```

9. Mendefinisikan Fungsi Frekuensi:

Membuat fungsi frek untuk menghitung jumlah data dalam setiap kelas interval.

```
> # Fungsi untuk menghitung frekuensi data dalam setiap kelas
> frek = function(x, y, z) {
+   a = 0
+   for (i in 1:n) {
+     if (x[i] >= y && x[i] <= z) {
+       a = a + 1
+     }
+   }
+   print(a)
+ }
```

10. Menghitung Frekuensi untuk Setiap Kelas:

Menggunakan fungsi `frek()` untuk menghitung jumlah data dalam setiap rentang .

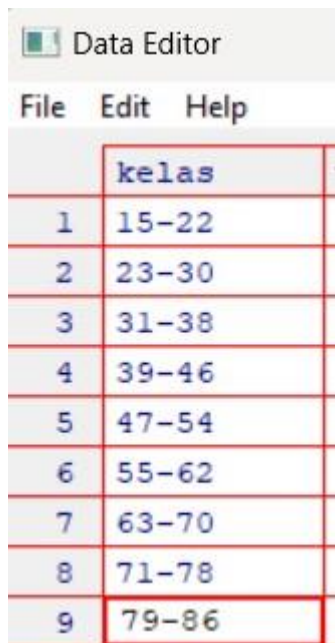
```
> #menghitung frekuensi setiap kelas
> f = frek(data,15,22)
[1] 3
> f = frek(data,23,30)
[1] 5
> f = frek(data,31,38)
[1] 4
> f = frek(data,39,46)
[1] 6
> f = frek(data,47,54)
[1] 0
> f = frek(data,55,62)
[1] 4
> f = frek(data,63,70)
[1] 6
> f = frek(data,71,78)
[1] 1
> f = frek(data,79,86)
[1] 1
```

11. Menyimpan Frekuensi Kelas secara Manual:

Menyimpan hasil frekuensi secara manual dalam vektor frekuensi.

```
> frekuensi <- c(3,5,4,6,0,4,6,1,1)
```

12. Membuat Tabel Data Frame untuk Frekuensi: Memasukkan vektor frekuensi ke dalam tabel.



	kelas
1	15-22
2	23-30
3	31-38
4	39-46
5	47-54
6	55-62
7	63-70
8	71-78
9	79-86

```

> frekuensi <- c(3,5,4,6,0,4,6,1,1)
> tabel = edit(data.frame())
> tabel
  kelas
1 15-22
2 23-30
3 31-38
4 39-46
5 47-54
6 55-62
7 63-70
8 71-78
9 79-86
> tabel$f <- frekuensi
> tabel
  kelas f
1 15-22 3
2 23-30 5
3 31-38 4
4 39-46 6
5 47-54 0
6 55-62 4
7 63-70 6
8 71-78 1
9 79-86 1

```

13. Membuat Tabel Frekuensi Menggunakan Package fdth::fdt

Menggunakan package fdth untuk membuat tabel frekuensi dengan perintah fdth.

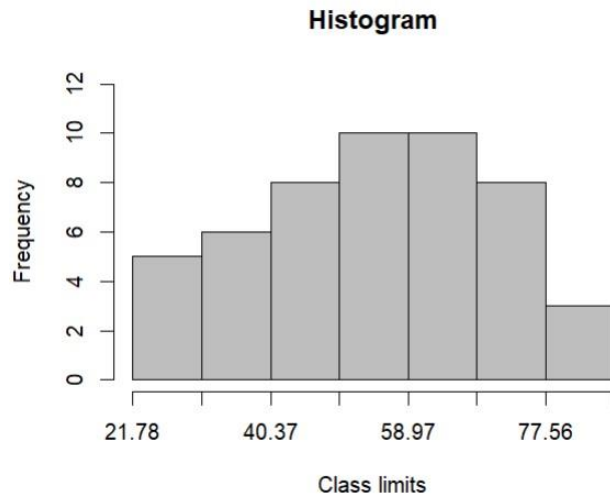
```

> # Membuat tabel frekuensi menggunakan package fdth
> tabelfrekuensi <- fdth::fdt(dataMahasiswa)
> tabelfrekuensi
  Class limits  f   rf rf(%)  cf cf(%)
[21.78,31.077)  5 0.10    10   5    10
[31.077,40.374)  6 0.12    12  11    22
[40.374,49.671)  8 0.16    16  19    38
[49.671,58.969) 10 0.20    20  29    58
[58.969,68.266) 10 0.20    20  39    78
[68.266,77.563)  8 0.16    16  47    94
[77.563,86.86)  3 0.06     6  50   100

```

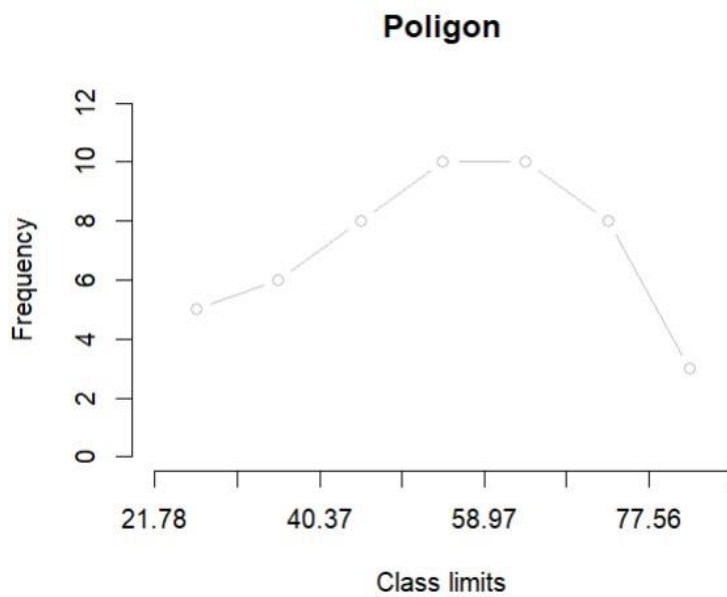
14. **Membuat Histogram:**

Menggunakan plot() untuk menggambar histogram dari tabel frekuensi.



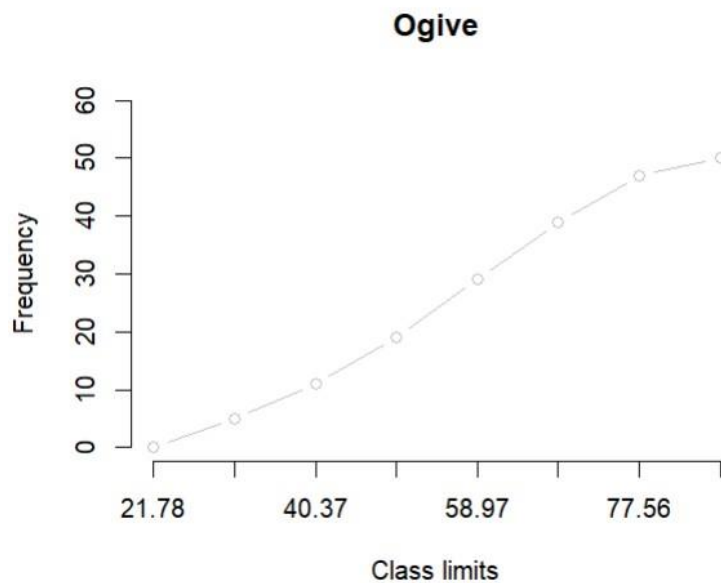
15. **Membuat Poligon Frekuensi:**

Menggunakan plot() dengan type='fp' untuk menggambar poligon frekuensi.



16. Membuat Ogive (Poligon Frekuensi Kumulatif):

Menggunakan `plot()` dengan `type='cfp'` untuk menggambar ogive (poligon kumulatif).



17. Menentukan ringkasan statistic menggunakan describe.

Describe membantu dalam memahami distribusi dan karakteristik umum dari data yang dianalisis.

```
> describe(data)
  vars  n  mean    sd median trimmed  mad min max range skew kurtosis  se
x1    1  30 45.53 18.65     40   45.46 22.24  15  80   65 0.16   -1.22 3.4
> |
```

D. Tugas

Latihan

Kelas ke-	Interval	Jumlah Frekuensi (F)
1	160 - 303	2
2	304 - 447	5
3	448 - 591	9
4	592 - 735	3
5	736 - 878	1

Carilah: Range, Varian, Standar Deviasi, Median dan Modus

1. Membuat dan menampilkan data frame.

```
> # Membuat data frame
> data <- data.frame(
+   kelas = c(1, 2, 3, 4, 5),
+   interval_bawah = c(160, 304, 448, 592, 736),
+   interval_atas = c(303, 447, 591, 735, 878),
+   frekuensi = c(2, 5, 9, 3, 1)
+ )
> # Menampilkan data frame
> print(data)
  kelas interval_bawah interval_atas frekuensi
1     1           160           303          2
2     2           304           447          5
3     3           448           591          9
4     4           592           735          3
5     5           736           878          1
> |
```

2. Menghitung nilai tengah.

```
> # Menghitung nilai tengah untuk setiap kelas
> data$nilai_tengah <- (data$interval_bawah + data$interval_atas) / 2
> |
```

3. Menghitung Range.

```
> # Menghitung Range
> range_data <- max(data$interval_atas) - min(data$interval_bawah)
> print(paste(range_data))
[1] "718"
> |
```

4. Menghitung Varian dan Standar Deviasi Populasi.


```

> # Hitung varian dan standar deviasi populasi
> n <- sum(data$frekuensi) # Total frekuensi
> varian <- sum(data$frekuensi * (data$nilai_tengah - mean_data)^2) / n
> standar_deviasi <- sqrt(varian)
> print(paste(varian))
[1] "19890.731875"
> print(paste(standar_deviasi))
[1] "141.034505972829"
~ |

```

5. Menghitung Median.

```

> # Menghitung Median dengan rumus  $Me = b + p * ((\frac{1}{2} * n) - F / f)$ 
> b_median <- data$interval_bawah[kelas_median]
> p <- data$interval_atas[1] - data$interval_bawah[1] # Panjang kelas
> F <- ifelse(kelas_median == 1, 0, frekuensi_kumulatif[kelas_median - 1]) # Frekuensi kumulatif sebelum kelas median
> f <- data$frekuensi[kelas_median] # Frekuensi kelas median
> n <- sum(data$frekuensi) # Total frekuensi
> median_new <- b_median + p * ((0.5 * n - F) / f)
> print(paste(median_new))
[1] "495.666666666667"
> |

```

6. Menghitung Modus.

```

> # Menghitung Modus dengan rumus  $Mo = b + p * (d_1 / (d_1 + d_2))$ 
> b_modus <- data$interval_bawah[kelas_modus]
> d1 <- data$frekuensi[kelas_modus] - ifelse(kelas_modus == 1, 0, data$frekuensi[kelas_modus - 1])
> d2 <- data$frekuensi[kelas_modus] - ifelse(kelas_modus == nrow(data), 0, data$frekuensi[kelas_modus + 1])
> modus_new <- b_modus + p * (d1 / (d1 + d2))
> print(paste(modus_new))
[1] "505.2"
~ |

```