

## PERTEMUAN 1: INTRO R DAN TABEL DISTRIBUSI FREKUENSI

### Contents

<b>1. Pengenalan R.....</b>	<b>2</b>
<b>1.1. Tipe dan Struktur Data .....</b>	<b>2</b>
<b>1.2. Operator Assignment .....</b>	<b>3</b>
<b>1.3. Operator Aritmetika .....</b>	<b>3</b>
<b>1.4. Operator Relasional .....</b>	<b>4</b>
<b>1.5. Operator Logika .....</b>	<b>4</b>
<b>2. Membuat Tabel Distribusi Frekuensi (TDF) di R.....</b>	<b>5</b>
<b>2.1. Membuat TDF Secara Manual .....</b>	<b>5</b>
<b>2.2. Pembuatan TDF di R.....</b>	<b>6</b>
<b>3. Memvisualisasi TDF dalam Bentuk Histogram dan Poligon .....</b>	<b>8</b>
<b>3.1. Membuat Histogram dan Poligon Secara Manual.....</b>	<b>8</b>
<b>3.2. Pembuatan Histogram dan Poligon di R .....</b>	<b>9</b>

## 1. Pengenalan R

### Apa itu statistika:

- Statistika adalah sebuah ilmu yang mempelajari bagaimana cara merencanakan, mengumpulkan, menganalisis, lalu menginterpretasikan, dan akhirnya mempresentasikan data. Singkatnya, statistika adalah ilmu yang bersangkutan dengan suatu data.
- Statistika digunakan untuk menemukan pola, menguji hipotesis, membuat prediksi, dan membantu pengambilan keputusan berdasarkan data.
- Dalam prakteknya, statistika dapat diterapkan di berbagai bidang seperti sains, bisnis, ekonomi, teknik, dan penelitian sosial.
- Ada berbagai jenis perangkat lunak yang digunakan untuk analisis statistika, di antaranya: R, Python, SPSS, MATLAB, dll. Namun, kita akan menggunakan R.

### Apa itu R-Programming?

R merupakan bahasa pemrograman yang dapat digunakan untuk pengolahan data atau pun untuk kalkulasi sebuah perhitungan aritmetika. Selain itu, R juga dapat digunakan untuk memvisualisasikan data yang sudah diolah dan berbagai hal yang berkaitan dengan matematika.



R adalah command-line interface (CLI) yang artinya pengguna harus mengetik perintah langsung di console R. Sedangkan RStudio adalah integrated development environment (IDE) yang dibuat untuk memperkaya pengalaman pengguna saat bekerja dengan R.

### Operator R

Di dalam R terdapat beberapa operator, antara lain:

- Operator Assignment
- Operator Aritmetika
- Operator Relasional / Perbandingan
- Operator Logika

#### 1.1. Tipe dan Struktur Data

Tipe Data	Contoh	Keterangan
-----------	--------	------------

Logical	TRUE, FALSE	Nilai Boolean
Numeric	12.3, 5, 999	Segala jenis angka
Integer	23L, 97L, 3L	Bilangan bulat
Complex	2i, 3i, 9i	Bilangan kompleks
Character	'a', 'b', "123"	Karakter dan string
Factor	1, 0, "Merah"	Dapat berupa numerik atau string (namun pada proses akan terbaca sebagai angka)
Raw	Identik dengan "hello"	Segala jenis data yang disimpan sebagai raw bytes

## 1.2.Operator Assignment

Operator assignment adalah operator yang digunakan untuk memberikan nilai kepada sebuah variabel. Dalam bahasa R, terdapat variasi operator assignment selain tanda sama dengan (=), tetapi ada juga `->` dan `<-` dengan sintaks:

`variabel <- nilai`

`nilai -> variabel`

Contoh:

```
> a <- "Halo,"
> "kawan" -> b
> c = 2
> d = "\n"
> cat(a, b, c, d)
Halo, kawan 2
```

## 1.3.Operator Aritmetika

Operator aritmetika adalah operator yang digunakan untuk melakukan operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan operasi lainnya pada data numerik.

Simbol	Keterangan
<code>+</code>	<i>Addition</i> , untuk operasi penjumlahan
<code>-</code>	<i>Subtraction</i> , untuk operasi pengurangan
<code>*</code>	<i>Multiplication</i> , untuk operasi perkalian
<code>/</code>	<i>Division</i> , untuk operasi pembagian
<code>^</code>	<i>Exponentiation</i> , untuk operasi pemangkatan
<code>%%</code>	<i>Modulus</i> , untuk mencari sisa pembagian
<code>%/%</code>	<i>Integer</i> , untuk mencari bilangan bulat hasil pembagian saja dan tanpa sisa pembagian

Contoh:

```
> 7+2      > 7*2      > 7^2
[1] 9        [1] 14        [1] 49
> 7-2      > 7/2      > 7%%2      > 7%/%2
[1] 5        [1] 3.5      [1] 1        [1] 3
```

## 1.4.Operator Relasional

Operator relasional (juga dikenal sebagai operator perbandingan) adalah operator yang digunakan untuk membandingkan dua nilai atau ekspresi dalam pemrograman. Hasil dari perbandingan ini biasanya berupa nilai boolean (True atau False), yang mengindikasikan apakah kondisi yang diuji itu benar atau tidak.

Simbol	Keterangan
>	Lebih besar dari
<	Lebih kecil dari
==	Sama dengan
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
!=	Tidak sama dengan

Contoh:

```
> a = 7
> b <- 2
> a > b
[1] TRUE
> a < b
[1] FALSE
> a >= b
[1] TRUE
> a <= b
[1] FALSE
> a != b
[1] TRUE
```

## 1.5.Operator Logika

Operator logika (atau operator boolean) adalah operator yang digunakan untuk mengoperasikan nilai boolean (True atau False) dalam pemrograman. Operator logika menggabungkan dua atau lebih kondisi atau ekspresi boolean untuk menghasilkan nilai boolean baru yang mewakili hasil dari operasi logika tersebut.

Simbol	Keterangan
&&	Operator logika AND
	Operator logika OR
!	Operator logika NOT
&	Operator logika AND bitwise
	Operator logika OR bitwise

Contoh:

```
> a = TRUE
> b = FALSE
> a && b
[1] FALSE
> a || b
[1] TRUE
> !a
[1] FALSE
> a & b
[1] FALSE
> a | b
[1] TRUE
> !a && b
[1] FALSE
> a <- c(TRUE, TRUE, FALSE, FALSE)
> a
[1] TRUE TRUE FALSE FALSE
> b <- c(TRUE, FALSE, TRUE, FALSE)
> b
[1] TRUE FALSE TRUE FALSE
> a & b
[1] TRUE FALSE FALSE FALSE
> a | b
[1] TRUE TRUE TRUE FALSE
> a && b
Error in a && b : 'length = 4' in coercion to 'logical(1)'
```

Perbedaan antara operator logika biasa (AND, OR) dan operator logika bitwise antara lain:

- Operator logika biasa: Hanya membandingkan elemen pertama dari dua vector dan mengembalikan nilai boolean tunggal (TRUE atau FALSE). Biasanya digunakan untuk pengujian logika skalar dalam kondisi if.
- Operator logika bitwise: Membandingkan setiap elemen dari dua vektor secara elemen per elemen, dan mengembalikan vektor logika dengan panjang yang sama. Ini digunakan untuk operasi perbandingan vektor atau matriks.

## 2. Membuat Tabel Distribusi Frekuensi (TDF) di R

Tabel Distribusi Frekuensi adalah alat statistik yang digunakan untuk menunjukkan berapa kali atau frekuensi setiap nilai atau kelompok nilai muncul dalam suatu data set.

Contoh Soal:

Diberikan data nilai ujian (dalam skala 0-100) dari 20 siswa sebagai berikut:

85, 90, 78, 93, 88, 74, 80, 85, 91, 87, 89, 84, 76, 98, 82, 77, 90, 85, 96, 81.

### 2.1. Membuat TDF Secara Manual

**Langkah 1:** Mengurutkan data

Urutkan data acak dari data terkecil hingga terbesar. Tujuannya agar range data diketahui dan mempermudah penghitungan frekuensi tiap kelas.

Data setelah diurutkan: 74, 76, 77, 78, 80, 81, 82, 84, 85, 85, 85, 87, 88, 89, 90, 90, 91, 93, 96, 98

**Langkah 2:** Menentukan banyak kelas

Kelas adalah pengelompokan data dengan interval dan rentang tertentu. Rumus kelas (Aturan Sturges):

$$k = 1 + 3,322 \log n$$

Pada contoh soal:

$$k = 1 + 3,322 \log 20$$

$$k = 1 + 4,322$$

$$k = 5,322 \approx 5$$

Banyak kelas dari data nilai ujian adalah 5.

### Langkah 3: Menentukan panjang (interval) kelas

Rumus:

$$i = \frac{r}{k}$$

Dengan r adalah range (selisih data terbesar dengan data terkecil) dan k adalah banyak kelas. Panjang kelas harus dibulatkan ke atas.

Pada contoh soal:

$$i = \frac{98 - 74}{5} = \frac{24}{5} = 4,8 \approx 5$$

Panjang kelas dari data nilai ujian adalah 5.

### Langkah 4: Menentukan frekuensi tiap kelas

74, 76, 77, 78, 80, 81, 82, 84, 85, 85, 85, 87, 88, 89, 90, 90, 91, 93, 96, 98

- 74 – 78: 4 data (74, 76, 77, 78)
- 79 – 83: 3 data (80, 81, 82)
- 84 – 88: 6 data (84, 85, 85, 85, 87, 88)
- 89 – 93: 5 data (89, 90, 90, 91, 93)
- 94 – 98: 2 data (96, 98)

### Langkah 5: Memasukkan data ke dalam TDF

	Kelas	Frekuensi
1	75 – 78	4
2	79 – 83	3
3	84 – 88	6
4	89 – 93	5
5	94 – 98	2

## 2.2.Pembuatan TDF di R

### Langkah 1: Mengurutkan data

```
> data = c(85, 90, 78, 93, 88, 74, 80, 85, 91, 87, 89, 84, 76, 98, 82, 77, 90, 85, 96, 81)
> dataSort = sort(data)
> dataSort
[1] 74 76 77 78 80 81 82 84 85 85 85 87 88 89 90 90 91 93 96 98
> n = length(dataSort)
> n
[1] 20
```

### Langkah 2: Menentukan banyak kelas

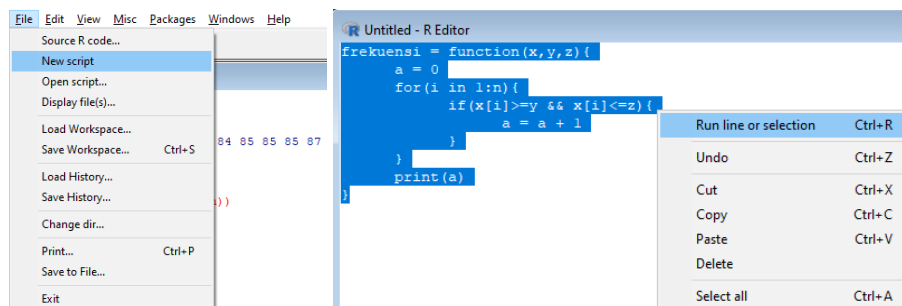
```
> K = 1 + (3.322 * log10(n))
> K
[1] 5.322022
> k = round(K)
> k
[1] 5
```

### Langkah 3: Menentukan panjang (interval) kelas

```
> nmax = max(dataSort)
> nmax
[1] 98
> nmin = min(dataSort)
> nmin
[1] 74
> r = nmax - nmin
> r
[1] 24
> I = r/k
> I
[1] 4.8
> i = ceiling(I)
> i
[1] 5
```

### Langkah 4: Menentukan frekuensi tiap kelas

Kita dapat menuliskan kode yang memiliki lebih dari satu baris dengan membuat script baru. Caranya pada menu bar, pilih File -> New script. Jika ingin merunning kode yang sudah diketik, blok kode tersebut, lalu tekan Ctrl+R.

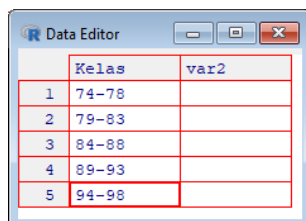


```
> frekuensi(dataSort, 74, 78)
[1] 4
> frekuensi(dataSort, 79, 83)
[1] 3
> frekuensi(dataSort, 84, 88)
[1] 6
> frekuensi(dataSort, 89, 93)
[1] 5
> frekuensi(dataSort, 94, 98)
[1] 2
> f = c(4, 3, 6, 5, 2)
> f
[1] 4 3 6 5 2
```

### Langkah 5: Memasukkan data ke dalam TDF

```
> TDF = edit(data.frame())
```

Ubah var1 menjadi Kelas dengan type character.



Setelah mengisi TDF, klik close (Jangan minimize, karena ditakutkan not responding).

```

> TDF
  Kelas
1 74-78
2 79-83
3 84-88
4 89-93
5 94-98

> TDF$Frekuensi = f
> TDF
  Kelas Frekuensi
1 74-78         4
2 79-83         3
3 84-88         6
4 89-93         5
5 94-98         2

```

### 3. Memvisualisasi TDF dalam Bentuk Histogram dan Poligon

Histogram adalah jenis grafik atau diagram batang yang digunakan untuk menggambarkan distribusi frekuensi atau distribusi data dalam bentuk interval tertentu. Histogram membantu kita memahami pola atau kecenderungan data, seperti sebaran, kepadatan, dan bentuk distribusinya.

Poligon histogram adalah bentuk grafik yang serupa dengan histogram, namun alih-alih menggunakan batang, kita menggunakan titik-titik yang dihubungkan dengan garis.

#### 3.1. Membuat Histogram dan Poligon Secara Manual

##### Langkah 1: Membuat Titik Tengah

Titik tengah ini akan digunakan sebagai posisi pada sumbu X untuk menggambar titik.

$$\text{Titik Tengah Kelas} = \frac{\text{Batas Bawah} + \text{Batas Atas}}{2}$$

$$(74 - 78) \rightarrow \text{Titik Tengah} = \frac{74 + 78}{2} = \frac{152}{2} = 76$$

$$(79 - 83) \rightarrow \text{Titik Tengah} = \frac{79 + 83}{2} = \frac{162}{2} = 81$$

$$(84 - 88) \rightarrow \text{Titik Tengah} = \frac{84 + 88}{2} = \frac{172}{2} = 86$$

$$(89 - 93) \rightarrow \text{Titik Tengah} = \frac{89 + 93}{2} = \frac{182}{2} = 91$$

$$(94 - 98) \rightarrow \text{Titik Tengah} = \frac{94 + 98}{2} = \frac{192}{2} = 96$$

##### Langkah 2: Membuat Histogram dan Poligon dari TDF

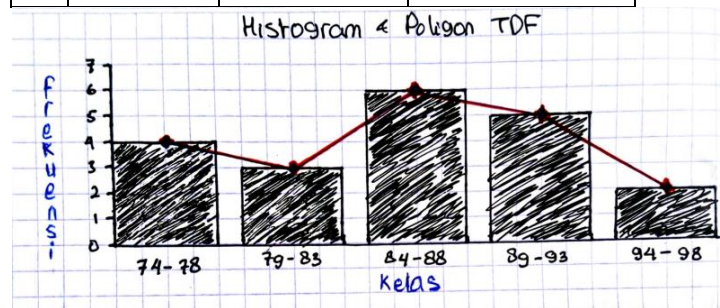
Beri judul histogram berdasarkan masalah. Beri nama label pada sumbu horizontal (x-axis) dengan “Kelas” dan sumbu vertikal (y-axis) dengan “Frekuensi”.

Gambarkan sebuah batang pada setiap kelas sesuai dengan frekuensinya.

	Kelas	Frekuensi	Titik Tengah
1	75 – 78	4	76
2	79 – 83	3	81
3	84 – 88	6	86



4	89 – 93	5	91
5	94 – 98	2	96



### 3.2.Pembuatan Histogram dan Poligon di R

#### Langkah 1: Membuat Titik Tengah

```
> # Membuat Titik Tengah pada TDF
> # Rumus TTK = (BB + BA) / 2
> (74 + 78) / 2
[1] 76
> (79 + 83) / 2
[1] 81
> (84 + 88) / 2
[1] 86
> (89 + 93) / 2
[1] 91
> (94 + 98) / 2
[1] 96
```

```
> TTK = c(76, 81, 86, 91, 96)
> TDF$Titik_tengah = TTK
> TDF
  Kelas Frekuensi Titik_tengah
1 74-78         4          76
2 79-83         3          81
3 84-88         6          86
4 89-93         5          91
5 94-98         2          96
```

#### Langkah 2: Membuat Histogram dan Poligon dari TDF

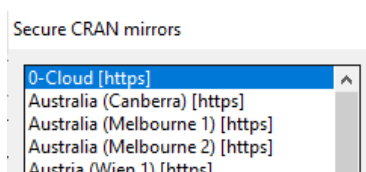
Data setelah diurutkan: 75, 76, 77, 78, 80, 81, 82, 84, 85, 85, 85, 85, 87, 88, 89, 90, 90, 91, 92, 93, 94

```
> library(ggplot2)
Error in library(ggplot2) : there is no package called 'ggplot2'
```

Jika menerima pesan kesalahan “there is no package called ‘ggplot2’” saat mencoba memuat library ggplot2, itu berarti bahwa paket ggplot2 belum terpasang di sistem. Untuk memperbaikinya, kita perlu menginstal terlebih dahulu paket ggplot2 (melalui konsol R biasa atau RStudio):

```
> install.packages("ggplot2")
Warning in install.packages("ggplot2") :
  'lib = "C:/Program Files/R/R-4.3.2/library"' is not writable
--- Please select a CRAN mirror for use in this session ---
```

Ini akan mengunduh dan menginstal ggplot2 dari CRAN (respositori R resmi). Pastikan komputer terhubung ke internet saat melakukan ini.



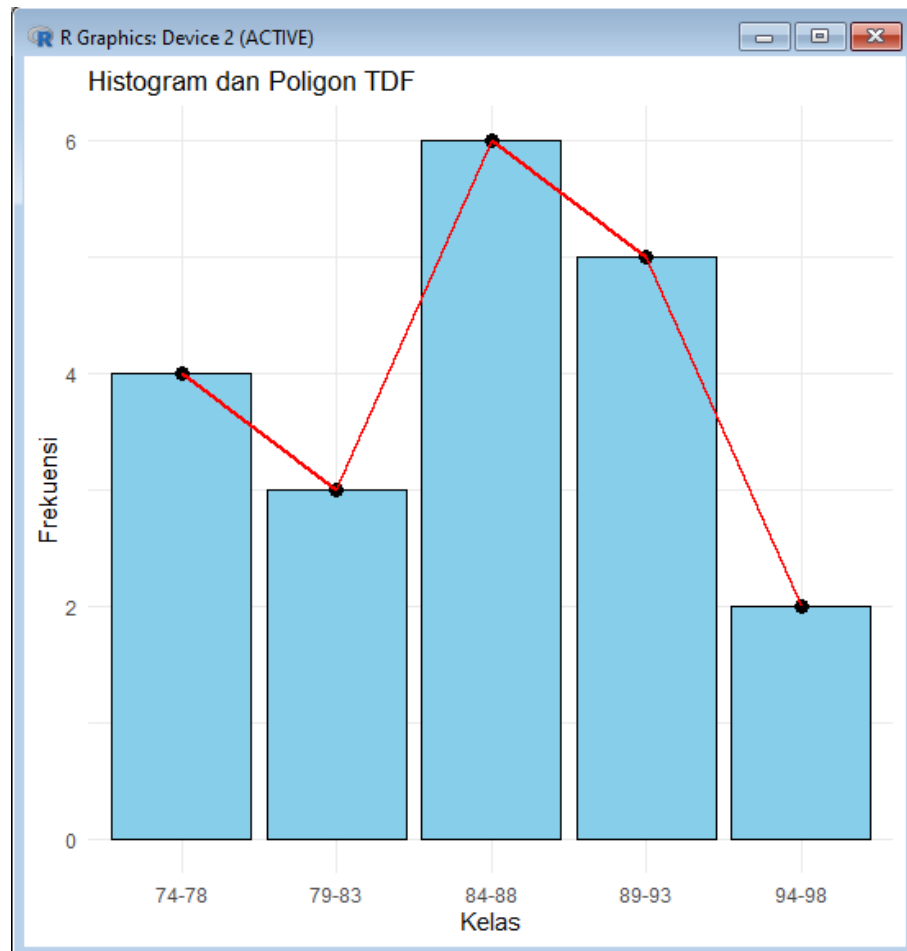
Saat memilih CRAN mirror, pilih 0-Cloud karena ini adalah mirror cloud resmi R yang dapat memilih server terbaik berdasarkan lokasi, memberikan kecepatan yang lebih optimal dan lebih stabil.

Namun, boleh juga memilih Indonesia (Banda Aceh) jika ingin mendapatkan kecepatan maksimal dari server local.

Setelah library ggplot2 terinstal, ketik:

```
> library(ggplot2)

# Buat plot dengan ggplot2
ggplot(TDF, aes(x = Kelas, y = Frekuensi)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  geom_point(aes(x = as.numeric(as.factor(Kelas)), y = Frekuensi), color = "black", size = 3) +
  geom_line(aes(x = as.numeric(as.factor(Kelas)), y = Frekuensi), color = "red", linetype = "solid", size = 1) +
  labs(title = "Histogram dan Poligon TDF", x = "Kelas", y = "Frekuensi") +
  theme_minimal()
```



Jika hanya ingin membuat histogram dan garis poligon saja (kelas, frekuensi, titik tengah kelas sudah diketahui):

```

Tabel_TDF <- data.frame(
  Kelas = c("74-78", "79-83", "84-88", "89-93", "94-98"),
  Frekuensi = c(4, 3, 6, 5, 2),
  Titik_tengah = c(76, 81, 86, 91, 96)
)

# Buat histogram dan garis poligon menggunakan library ggplot2
library(ggplot2)
# Buat plot dengan ggplot2
ggplot(Tabel_TDF, aes(x = Kelas, y = Frekuensi)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  geom_point(aes(x = as.numeric(as.factor(Kelas)), y = Frekuensi), color = "black", size = 3) +
  geom_line(aes(x = as.numeric(as.factor(Kelas)), y = Frekuensi), color = "red", linetype = "solid", size = 1) +
  labs(title = "Histogram dan Poligon TDF", x = "Kelas", y = "Frekuensi") +
  theme_minimal()

```