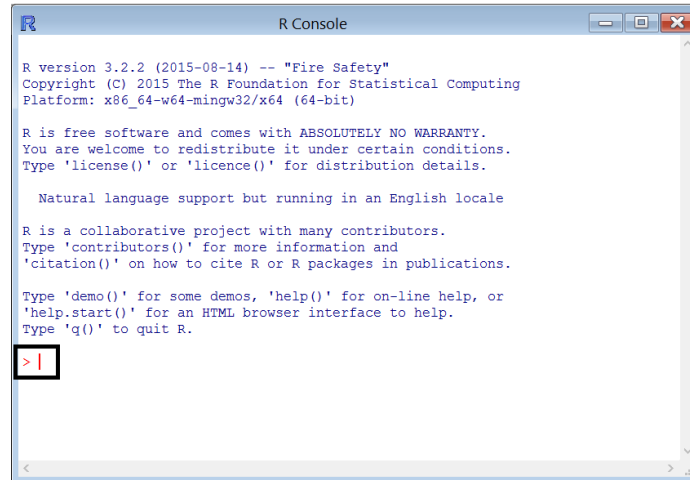


MEMULAI R

- Ketika kita memulai program R, akan muncul tampilan seperti Gambar 1



Gambar 1 R console

- Tanda > merupakan **command prompt** yang menunggu user untuk mengetikkan perintah.

Menulis perintah pada R console

- Misal ingin dihitung $10+20$, maka user dapat mengetikkan secara langsung

```
> 10+20  
[1] 30
```

Contoh lain misalkan diberikan x, y dan z,

```
x<-6  
y<-4  
z<-x+y
```

Objek z dapat dipanggil kembali dengan perintah

```
z  
[1] 10
```

Untuk membuat daftar objek x, y dan z,

```
> ls()  
[1] "x"          "y"          "z"
```

- *Command* untuk menghilangkan objek adalah rm, contohnya:

```
> rm(x,y)
```

- Berikut adalah beberapa fungsi aritmatika

Operasi	Operator	Contoh Input	Contoh Output
Penjumlahan	+	10+2	12
Pengurangan	-	9-3	6
Perkalian	*	5*5	25
Pembagian	/	10/3	3
Pangkat	^	5^2	25

- R mengenal spasi ,misal

```
> 10+ 20
[1] 30
```

- Contoh operasi aritmatika lain:

```
> 3*3*3*3
[1] 81
> 45^2
[1] 2025
> 1+2*3
[1] 7
> 34-4*2
[1] 26
> sqrt(81)
[1] 9
> abs(-9)
[1] 9
```

Menyimpan data sebagai suatu variabel

- R dapat menyimpan data sebagai suatu variabel. Misal kita ingin mengenakan nilai 350 pada variabel dengan nama sales,

```
> sales<-350
> sales
[1] 350
```

- Misal royalti yang didapat adalah 7 dan ada fungsi **revenue** yang merupakan perkalian sales dan royalti,

```
> royalty<-7
> revenue<-sales*royalty
> revenue
[1] 2450
```

- Fungsi **round** untuk membulatkan angka decimal,

```
> round(3.1415)
[1] 3
> round(3.14165,2)
[1] 3.14
```

Menyimpan data sebagai suatu vektor

- Variabel dalam R yang dapat menyimpan multivariat data disebut dengan **vector**
- Misal akan disimpan data 0, 100, 200, 50, 0, 0, 0, 0, 0, 0, 0, 0 maka kita dapat menyimpan data tersebut dengan fungsi **c()**

- Variabel tunggal misal dinamakan sales.bulan maka,

```
> sales.bulan<-c(0, 100, 200, 50, 0, 0, 0, 0, 0, 0, 0, 0)
> sales.bulan
[1] 0 100 200 50 0 0 0 0 0 0 0 0
```

- Jika user ingin memanggil data kedua maka,

```
> sales.bulan[2]
[1] 100
```

- Jika 100 tadi ingin dinamakan sebagai variabel baru, maka,

```
> datakedua<-sales.bulan[2]
> datakedua
[1] 100
```

- Misalkan data kelima ternyata salah, yang benar adalah 25 maka,

```
> sales.bulan[5]<-25
> sales.bulan
[1] 0 100 200 50 25 0 0 0 0 0 0 0
```

- Fungsi **length()** digunakan untuk mengetahui banyaknya elemen dalam vector

```
> length(x=sales.bulan)
[1] 12
```

- Apabila user ingin mengalikan data pada sales.bulan dengan 10. Artinya semua elemen dalam variable dikalikan 10. Hal ini analog dengan penjumlahan, pengurangan, pembagian dan pangkat

```
> sales.bulan*10
[1]      0 1000 2000   500   250      0      0      0      0      0      0
```

- Misal akan didefinisikan vector baru,

```
> data.baru<-c(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
```

- Dan didefinisikan variabel $\text{untung} = \text{sales.bulan} * 7$,

```
> untung<-sales.bulan*7
> untung
[1]      0  700 1400  350  175      0      0      0      0      0      0
```

- Selanjutnya, dicari pembagian untung dengan data baru, sehingga diperoleh

```
> untung/data.baru
[1] 0.000000 25.000000 45.161290 11.666667 5.645161 0.000000 0.000000
[8] 0.000000 0.000000 0.000000 0.000000 0.000000
```

- *Command* untuk mengulangi vector adalah rep, contohnya:

[illegible]

```
> rep(1:3,6)
[1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
> rep(1:3,c(6,6,6))
[1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3
```

Contoh operasi sederhana dalam R

```
> x<-c(6,8,9)
> y<-c(1,2,4)
> x+y
[1] 7 10 13
```

```
> x*y
[1]  6 16 36
```

```
> x<-c(6,8,9)
> x+2
[1] 8 10 11
```

Menyimpan data berupa Text

- R juga dapat membuat data text,

```
> salam<-"Assalamualaikum"
> salam
[1] "Assalamualaikum"
```

- Analog dengan data numeric, data text juga dapat menggunakan fungsi c() dan memanggilnya kembali

```
> Hari<-c("Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Ahad")
> Hari
[1] "Senin" "Selasa" "Rabu" "Kamis" "Jumat" "Sabtu" "Ahad"
> Hari[7]
[1] "Ahad"
```

- Dari contoh diatas, nampak bahwa karakter vector terdiri dari 7 elemen
- Fungsi `nchar()` digunakan untuk menghitung jumlah karakter dari elemen yang telah didefinisikan oleh user

```
> nchar(x=Hari)
[1] 5 6 4 5 5 5 4
```

Menyimpan data True or False

- Operator `==` dapat digunakan dalam R untuk menuliskan true or false, contoh

```
> 3+3==6
[1] TRUE
> 3+3==7
[1] FALSE
```

- Beberapa operasi logika

Operasi	Operator	Contoh Input	Jawaban
Kurang dari	<	2<3	TRUE
Kurang dari atau sama dengan	<=	2<=2	TRUE
Lebih besar dari	>	2>3	FALSE
Lebih besar dari atau sama dengan	>=	2>=2	TRUE
Sama dengan	==	2==3	FALSE
Tidak sama dengan	!=	2!=3	TRUE
Tida	!	!(1==1)	FALSE
Atau		(1==1) (2==3)	TRUE
dan	&	(1==1)&(2==3)	FALSE

-Contoh

```
> 100<90
[1] FALSE
> 90<100
[1] TRUE
> sqrt(25)==5
[1] TRUE
> (2+2 == 4) | (2+2 == 5)
[1] TRUE
> (2+2 == 4) & (2+2 == 5)
[1] FALSE
> ! (2+2 == 5)
[1] TRUE
> 2+2 != 5
[1] TRUE
```

Vector dari logika

-Sama seperti penyimpanan data numeric maupun text, fungsi c() juga dapat digunakan

```
> x<-c(TRUE, TRUE, FALSE)
> x
[1] TRUE TRUE FALSE
```

-Misal menggunakan data sales.bulan, user ingin mengetahui sales perbulannya,

```
> sales.bulan>0
[1] FALSE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

-Contoh lain

```
> "kucing"=="kucing"
[1] TRUE
> " kucing"=="kucing"
[1] FALSE
```

Mengekstrak multiple elemen

-Contoh

```
> sales.bulan[c(1,2)]
[1] 0 100
> Hari[c(5,7)]
[1] "Jumat" "Ahad"
```

-Atau penjualan dari bulan ke 1 sampai dengan 5,

```
> sales.bulan[1:5]  
[1] 0 100 200 50 25
```

-Penjualan yang besarannya >0,

```
> sales.bulan[sales.bulan>0]  
[1] 100 200 50 25
```

Quiting R

-Jika user ingin log out dari R maka dapat menggunakan fungsi q()

