

# Konsep - konsep pemrograman lanjut

---

## Fitur - fitur *built-in*

`seq()`: Mendefinisikan sikuen

```
seq(0,10, by=2)
```

1. 0
2. 2
3. 4
4. 6
5. 8
6. 10

```
seq(0,100, by = 10)
```

1. 0
2. 10
3. 20
4. 30
5. 40
6. 50
7. 60
8. 70
9. 80
10. 90
11. 100

```
seq(0,30, by = 2)
```

1. 0
2. 2
3. 4
4. 6
5. 8
6. 10

7. 12
8. 14
9. 16
10. 18
11. 20
12. 22
13. 24
14. 26
15. 28
16. 30

`sort()` : Mengurutkan vektor

```
v <- c(2, 7, 1, 49, 54, 32)
v
```

1. 2
2. 7
3. 1
4. 49
5. 54
6. 32

```
sort(v) # dari kecil ke besar
```

1. 1
2. 2
3. 7
4. 32
5. 49
6. 54

```
sort(v, decreasing = T) # dari besar ke kecil
```

1. 54
2. 49
3. 32
4. 7
5. 2
6. 1

```
nama <- c('s', 'a', 'n', 'd', 'y')
nama
```

1. 's'
2. 'a'
3. 'n'
4. 'd'
5. 'y'

```
sort(nama)
```

1. 'a'
2. 'd'
3. 'n'
4. 's'
5. 'y'

```
nama <- c('s', 'a', 'n', 'd', 'Y')
sort(nama)
```

1. 'a'
2. 'd'
3. 'n'
4. 's'
5. 'Y'

```
nama <- c('s', 'a', 'n', 'd', 'Y', 'A')
sort(nama)
```

1. 'a'
2. 'A'
3. 'd'
4. 'n'
5. 's'
6. 'Y'

`rev()` : Membalikan elemen di dalam suatu objek

```
b <- seq(1, 10)
b
```

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
```

```
rev(b)
```

```
1. 10
2. 9
3. 8
4. 7
5. 6
6. 5
7. 4
8. 3
9. 2
10. 1
```

```
d <- c('a', 'b', 'e', 'd')
d
```

```
1. 'a'
2. 'b'
3. 'e'
4. 'd'
```

```
rev(d)
```

1. 'd'
2. 'e'
3. 'b'
4. 'a'

**str()**: Menunjukkan struktur dari suatu objek

```
str(b)
```

```
int [1:10] 1 2 3 4 5 6 7 8 9 10
```

```
str(mtcars)
```

```
'data.frame':  32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8
19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92
3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
summary(mtcars)
```

	mpg	cyl	disp	hp
Min.	:10.40	Min. :4.000	Min. : 71.1	Min. :
				52.0
1st Qu.:	15.43	1st Qu.:4.000	1st Qu.:120.8	1st Qu.:
				96.5
Median	:19.20	Median :6.000	Median :196.3	Median
				:123.0

Mean	:20.09	Mean	:6.188	Mean	:230.7	Mean	
	:146.7						
3rd Qu.:	22.80	3rd Qu.:	8.000	3rd Qu.:	326.0	3rd	
Qu.:	180.0						
Max.	:33.90	Max.	:8.000	Max.	:472.0	Max.	
	:335.0						
	drat		wt		qsec		vs
Min.	:2.760	Min.	:1.513	Min.	:14.50	Min.	
	:0.0000						
1st Qu.:	3.080	1st Qu.:	2.581	1st Qu.:	16.89	1st	
Qu.:	0.0000						
Median	:3.695	Median	:3.325	Median	:17.71	Median	
	:0.0000						
Mean	:3.597	Mean	:3.217	Mean	:17.85	Mean	
	:0.4375						
3rd Qu.:	3.920	3rd Qu.:	3.610	3rd Qu.:	18.90	3rd	
Qu.:	1.0000						
Max.	:4.930	Max.	:5.424	Max.	:22.90	Max.	
	:1.0000						
	am		gear		carb		
Min.	:0.0000	Min.	:3.000	Min.	:1.000		
1st Qu.:	0.0000	1st Qu.:	3.000	1st Qu.:	2.000		
Median	:0.0000	Median	:4.000	Median	:2.000		
Mean	:0.4062	Mean	:3.688	Mean	:2.812		
3rd Qu.:	1.0000	3rd Qu.:	4.000	3rd Qu.:	4.000		
Max.	:1.0000	Max.	:5.000	Max.	:8.000		

## append() : Menggabungkan objek

```
v1 <- seq(1,5)
v2 <- seq(10,30, by=10)
```

```
append(v1,v2)
```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 10
7. 20
8. 30

## Memeriksa dan mengonversi tipe data pada objek - objek R

```
v1
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
is.vector(v1)
```

TRUE

```
is.data.frame(v1)
```

FALSE

```
is.data.frame(mtcars)
```

TRUE

```
as.list(v1)
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
as.matrix(v1)
```

- 1
- 2
- 3
- 4
- 5

## Fungsi - fungsi `apply`

```
v <- seq(10, 50, by=10)
v
```

1. 10
2. 20
3. 30
4. 40
5. 50

```
sample(v, 2) # mengambil dua buah sampel acak dari vektor
```

1. 30
2. 20

```
sample(1:100, 5)
```

1. 98
2. 17
3. 78
4. 100
5. 40

```
v <- 1:5
v
```

1. 1
2. 2
3. 3
4. 4
5. 5



```
tambah_acak <- function(x){  
  acak <- sample(1:100,1)  
  return(x + acak)  
}
```

```
tambah_acak(10)
```

63

```
hasil <- tambah_acak(20)  
hasil
```

91

**lapply(): dalam bentuk list**

```
lapply(v, tambah_acak)  
# outputnya dalam bentuk list
```

```
1. 80  
2. 95  
3. 36  
4. 102  
5. 8
```

**sapply(): dalam bentuk vektor**

```
sapply(v, tambah_acak)
```

```
1. 52  
2. 8  
3. 84  
4. 22  
5. 8
```

```
v1 <- seq(5,25, by=5)  
kuadrat <- function(bil){  
  return(bil^2)  
}
```

```
kuadrat(5)
```

25

```
lapply(v, kuadrat)
```

```
1. 1
2. 4
3. 9
4. 16
5. 25
```

```
sapply(v, kuadrat)
```

```
1. 1
2. 4
3. 9
4. 16
5. 25
```

## Fungsi anonim

```
v
```

```
1. 1
2. 2
3. 3
4. 4
5. 5
```

```
kuadrat <- function(bil){
  return(bil^2)
}
```

```
sapply(v, function(bil){bil^2}) # fungsi anonim
```

1. 1
2. 4
3. 9
4. 16
5. 25

Fungsi `apply` dengan banyak *input*

```
v
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
tambah_dua_bil <- function(b1,b2){  
  return(b1+b2)  
}
```

```
tambah_dua_bil(20,30)
```

50

```
sapply(v, tambah_dua_bil) # error
```

```
Error in FUN(X[[i]], ...): argument "b2" is missing, with  
no default  
Traceback:
```

1. `sapply(v, tambah_dua_bil)`
2. `lapply(X = X, FUN = FUN, ...)`
3. `FUN(X[[i]], ...)`

```
sapply(v, tambah_dua_bil, b2 = 10)
```

1. 11
2. 12
3. 13
4. 14
5. 15

## Ekspresi regular : RegEx

```
txt <- "Halo semuanya! Selamat Pagi! Cuaca lagi bagus, nih  
buat touring."
```

```
txt
```

'Halo semuanya! Selamat Pagi! Cuaca lagi bagus, nih buat touring.'

```
grepl("Halo",txt) # kata "Halo" ada di txt
```

TRUE

```
grepl("Malam", txt)
```

FALSE

```
grepl("halo", txt) # Sifatnya case-sensitive
```

FALSE

```
v <- c('a','d','k', 'l','t','k')  
grepl('k',v)
```

1. FALSE
2. FALSE
3. TRUE
4. FALSE
5. FALSE
6. TRUE

```
grep('k',v) # outputnya indeks
```

1. 3
2. 6

```
grep('a',v)
```

1

## Fungsi - fungsi matematika

`abs()` : menghitung nilai absolut

```
abs(-2)
```

2

```
v <- c(-3, -5, 7, 10)  
abs(v)
```

1. 3
2. 5
3. 7
4. 10

`sum()` : menghitung penjumlahan seluruh elemen

```
sum(2, 4, 6)
```

12

```
v <- c(2, 3, 4, 5)  
sum(v)
```

14

`mean()` : menghitung rata - rata aritmatika

```
mean(v)
```

3.5

```
mean(c(3,4,5))
```

4

`round()` : membulatkan nilai

```
round(2.777645)
```

3

```
round(2.777645, digits=2)
```

2.78

```
round(2.777645, 4)
```

2.7776

## ***Dates dan Timestamps***

```
Sys.Date() # waktu saat ini
```

2020-07-01

```
d <- Sys.Date()  
d
```

2020-07-01

```
class(d)
```

'Date'

```
d <- '1993-03-13'  
d
```

'1993-03-13'

```
class(d)
```

'character'

```
# dikonversi menjadi date  
b.day <- as.Date(d)  
b.day
```

1993-03-13

```
class(b.day)
```

'Date'

```
as.Date('Mar-13-93') # format tidak sesuai
```

```
Error in charToDate(x): character string is not in a  
standard unambiguous format  
Traceback:
```

```
1. as.Date("Mar-13-93")

2. as.Date.character("Mar-13-93")

3. charToDate(x)

4. stop("character string is not in a standard unambiguous
format")
```

```
as.Date('Mar-13-93', format = '%b-%d-%y')
```

1993-03-13

- `%d`: hari (desimal)
- `%m`: bulan (desimal)
- `%b`: bulan (singkatan)
- `%B`: bulan (tidak disingkat)
- `%y`: tahun (2 digit)
- `%Y`: tahun (4 digit)

```
as.Date('March,01,2009', format= "%B, %d, %Y")
```

2009-03-01

```
# POSIXct
```

```
as.POSIXct('11:03:05', format='%H:%M:%S')
```

```
[1] "2020-07-01 11:03:05 WIB"
```

```
strptime('11:03:05', format = '%H:%M:%S') # lebih banyak
dipakai di pemrograman R
```

```
[1] "2020-07-01 11:03:05 WIB"
```