

STK372 Komputasi Statistik 2

Pemrograman Berorientasi Objek dalam R

Class System S4

Dr. Agus M Soleh



IPB University
— Bogor Indonesia —

agusms@apps.ipb.ac.id

Outline

- 1 Objek: Class System S4
- 2 Konstruktor dan Aksesori
- 3 Fungsi Generik
- 4 Pewarisan Class
- 5 Transformasi dengan Matematik
- 6 Operasi Aritmetika
- 7 Method Subset
- 8 Pemeriksaan Suatu Class Objek

Objek: Class System S4

- Class System S4: Mengatasi masalah dalam Class System S3 dengan sistem objek lebih formal
- Salah satu keuntungan: sistem penurunan dari class/objek
- Dalam sistem objek formal: setiap objek didefinisikan secara formal dalam suatu class
- Sebuah class terdiri dari slot dengan tipe atau class spesifik
- Class dideklarasikan dengan fungsi setClass

```
> setClass("coords",  
+         representation(x = "numeric",  
+                        y = "numeric"))
```

- Berikut adalah definisi ulang class sebelumnya ke Class System S4

Konstruktor

Konstruktor:

```
> coords <- function(x, y) {
+   if (length(x) != length(y))
+     stop("length x dan y harus bernilai sama")
+   if (!is.numeric(x) || !is.numeric(y))
+     stop("x dan y harus vektor numeric")
+   new("coords", x = as.vector(x),
+       y = as.vector(y))
+ }
> pts <- coords(round(rnorm(5), 2),
+               round(rnorm(5), 2))
> pts
An object of class "coords"
Slot "x":
[1] -0.63  0.18 -0.84  1.60  0.33

Slot "y":
[1] -0.82  0.49  0.74  0.58 -0.31
```

Aksesori

Akses terhadap slot menggunakan fungsi slot atau operator @

```
> slot(pts, "x") # atau menggunakan operator @ --> pts@x
[1] -0.63  0.18 -0.84  1.60  0.33
> slot(pts, "y") # atau menggunakan operator @ --> pts@y
[1] -0.82  0.49  0.74  0.58 -0.31
```

Tetapi disarankan dengan 2 fungsi seperti sebelumnya:

```
> xcoords <- function(obj) obj@x
> ycoords <- function(obj) obj@y
> xcoords(pts)
[1] -0.63  0.18 -0.84  1.60  0.33
> ycoords(pts)
[1] -0.82  0.49  0.74  0.58 -0.31
```

Fungsi Generik

- Fungsi generik `show` setara dengan fungsi generik `print` pada class `System S3`
- Penciptaan fungsi generik menggunakan fungsi `setMethod`
- Argumen didefinisikan dalam signature

```
> setMethod(show, signature(object = "coords"),
+           function(object)
+             print(paste("(",
+                         format(xcoords(object)),
+                         ", ",
+                         format(ycoords(object)),
+                         ") ", sep=""),
+                 quote=FALSE))
> pts
[1] (-0.63, -0.82) ( 0.18,  0.49) (-0.84,  0.74)
[4] ( 1.60,  0.58) ( 0.33, -0.31)
```

Definisi Fungsi Generik Baru

- Mendefinisikan fungsi baru sebagai fungsi generik menggunakan:

```

> setGeneric("bbox",
+           function(obj)
+             standardGeneric("bbox"))
[1] "bbox"
> setMethod("bbox", signature(obj = "coords"),
+           function(obj)
+             matrix(c(range(xcoords(obj)),
+                       range(ycoords(obj))),
+                   nc = 2,
+                   dimnames = list(
+                     c("min", "max"),
+                     c("x:", "y:"))
+           ))
> bbox(pts)
x:      y:
min -0.84 -0.82
max  1.60  0.74

```

Fungsi Generik Plot

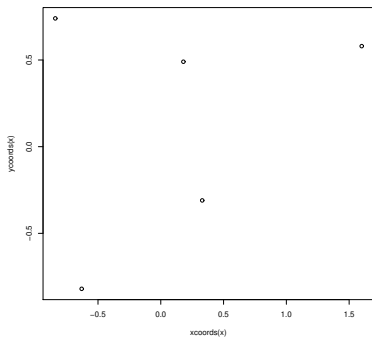
```

> setMethod("plot", signature(x="coords"),
+         function(x, bbox=FALSE, ...){
+           if (bbox) {
+             plot(xcoords(x), ycoords(x), ...);
+             x.1 <- c(bbox(x)[1], bbox(x)[2], bbox(x)[2], bbox(x)
+             x)[1]);
+             y.1 <- c(bbox(x)[3], bbox(x)[3], bbox(x)[4], bbox(x)
+             x)[4]);
+             polygon(x.1, y.1)
+           } else {
+             plot(xcoords(x), ycoords(x), ...)
+           }
+         })

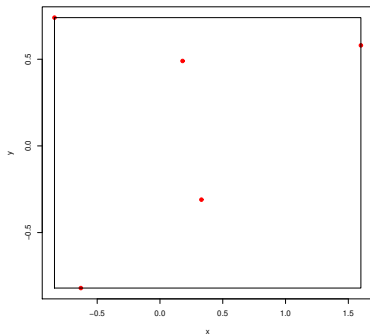
```


Fungsi Generik Plot

```
plot(pts)
```



```
plot(pts, bbox=T, pch=19, col="red", xlab="x", ylab="y")
```



Pewarisan Class

Terdapat class baru yang diturunkan dari coords dengan menambahkan slot nilai

```
> setClass("vcoords",
+         representation(nilai = "numeric"),
+         contains = "coords")
> vcoords <- function(x, y, nilai){
+   if ((length(x) != length(y)) ||
+       (length(x) != length(nilai)))
+     stop("length x, y, dan nilai harus bernilai sama")
+   if (!is.numeric(x) || !is.numeric(y)
+       || !is.numeric(nilai))
+     stop("x, y, dan nilai harus vektor numeric")
+   new("vcoords", x = as.vector(x),
+       y = as.vector(y),
+       nilai = as.vector(nilai))
+ }
> nilai <- function(obj) obj@nilai
```

Pewarisan Class

```
> vpts <- vcoords(xcoords(pts), ycoords(pts), round(100*runif
  (5)))
> vpts
[1] (-0.63, -0.82) ( 0.18,  0.49) (-0.84,  0.74)
[4] ( 1.60,  0.58) ( 0.33, -0.31)
```

Metode show yang diwariskan perlu didefinisi ulang

```
> setMethod(show, signature(object = "vcoords"),
+           function(object)
+             print(paste("(",
+                         format(xcoords(object)),
+                         ", ",
+                         format(ycoords(object)),
+                         "; ", format(nilai(object)),
+                         ") ", sep=""),
+                 quote=FALSE))
> vpts
[1] (-0.63, -0.82; 27) ( 0.18,  0.49; 37) (-0.84,  0.74; 57)
[4] ( 1.60,  0.58; 91) ( 0.33, -0.31; 20)
```

Pewarisan Class

Demikian juga method plot perlu didefinisi ulang

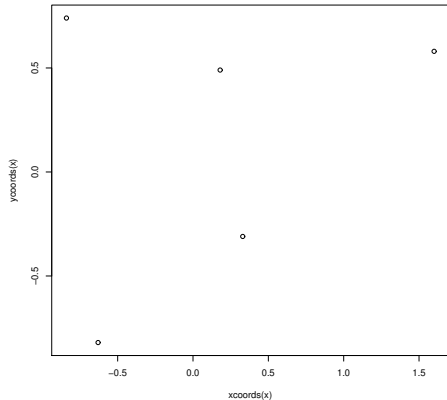
```
> setMethod("plot", signature(x="vcoords"),
+   function(x, txt=FALSE, bbox=FALSE, ...){
+     if (bbox) {
+       if (!txt) {
+         plot(xcoords(x), ycoords(x), ...);
+       } else {
+         plot(xcoords(x), ycoords(x), type="n", ...);
+         text(xcoords(x), ycoords(x), nilai(x), ...);
+       }
+     }
+     x.1 <- c(bbox(x)[1], bbox(x)[2], bbox(x)[2], bbox(x)[1]);
+     y.1 <- c(bbox(x)[3], bbox(x)[3], bbox(x)[4], bbox(x)[4]);
+     polygon(x.1, y.1)
+   } else {
```

Pewarisan Class

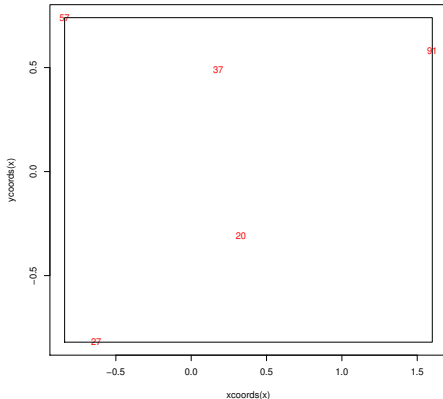
```
+     if (!txt) {  
+         plot(xcoords(x),ycoords(x), ...);  
+     } else {  
+         plot(xcoords(x),ycoords(x), type="n", ...);  
+         text(xcoords(x),ycoords(x), nilai(x), ...);  
+     }  
+ }  
+ })
```

Pewarisan Class

```
> plot(vpts)
```



```
> plot(vpts, txt=T, bbox=T,
      pch=19, col="red")
```



Tranformasi dengan Matematik

```
> setMethod("cos", signature(x = "vcoords"),  
+           function(x)  
+             vcoords(xcoords(x),  
+                     ycoords(x),  
+                     cos(nilai(x))))  
> cos(vpts)  
[1] (-0.63, -0.82; -0.2921388)  
[2] ( 0.18,  0.49;  0.7654141)  
[3] (-0.84,  0.74;  0.8998668)  
[4] ( 1.60,  0.58; -0.9943675)  
[5] ( 0.33, -0.31;  0.4080821)
```

Tranformasi dengan Matematik

Seperti pada Class System S3, terdapat group Math

```
> setMethod("Math", signature(x = "vcoords"),
+           function(x)
+             vcoords(xcoords(x),
+                   ycoords(x),
+                   callGeneric(nilai(x))))
> sin(vpts)
[1] (-0.63, -0.82; 0.9563759) ( 0.18, 0.49; -0.6435381)
[3] (-0.84, 0.74; 0.4361648) ( 1.60, 0.58; 0.1059875)
[5] ( 0.33, -0.31; 0.9129453)
> sqrt(vpts)
[1] (-0.63, -0.82; 5.196152) ( 0.18, 0.49; 6.082763)
[3] (-0.84, 0.74; 7.549834) ( 1.60, 0.58; 9.539392)
[5] ( 0.33, -0.31; 4.472136)
```


Operasi Aritmetika

```
> sameloc <- function(e1, e2) (  
+   length(nilai(e1)) == length(nilai(e2))  
+   || any(xcoords(e1) == xcoords(e2))  
+   || any(ycoords(e1) == ycoords(e2))  
+ )  
>  
> setMethod("Arith", signature(e1 = "vcoords",  
+                               e2 = "vcoords"),  
+   function(e1, e2){  
+     if (!sameloc(e1, e2))  
+       stop("Dibutuhkan titik identik")  
+     vcoords(xcoords(e1),  
+            ycoords(e2),  
+            callGeneric(nilai(e1),  
+                        nilai(e2)))  
+   })
```

Operasi Aritmetika

```

> vpts
[1] (-0.63, -0.82; 27) ( 0.18,  0.49; 37)
[3] (-0.84,  0.74; 57) ( 1.60,  0.58; 91)
[5] ( 0.33, -0.31; 20)
> vpts+vpts
[1] (-0.63, -0.82;  54) ( 0.18,  0.49;  74)
[3] (-0.84,  0.74; 114) ( 1.60,  0.58; 182)
[5] ( 0.33, -0.31;  40)
> 2+vpts
Error in 2 + vpts : non-numeric argument to binary operator

```

Berbeda dengan Class System S3, di S4 cukup menambah `setMethod("Arith", ..)` untuk operasi dengan numeric.

Operasi Aritmetika

Menambah operasi aritmetika dengan awal numeric

```
> setMethod("Arith",
+           signature(e1 = "numeric",
+                   e2 = "vcoords"),
+           function(e1, e2){
+             if (length(e1) > length(nilai(e2)))
+               stop("length yang tidak benar")
+             vcoords(xcoords(e2),
+                   ycoords(e2),
+                   callGeneric(as.vector(e1),
+                               nilai(e2)))
+           })
>
> 2*vpts
[1] (-0.63, -0.82; 54) ( 0.18, 0.49; 74)
[3] (-0.84, 0.74; 114) ( 1.60, 0.58; 182)
[5] ( 0.33, -0.31; 40)
```

Operasi Aritmetika

Menambah operasi aritmetika dengan akhir numeric

```

> setMethod("Arith", signature(e1 = "vcoords",
+                               e2 = "numeric"),
+           function(e1, e2){
+             if (length(nilai(e1)) < length(e2))
+               stop("length yang tidak benar")
+             vcoords(xcoords(e1),
+                     ycoords(e1),
+                     callGeneric(nilai(e1),
+                                 as.vector(e2)))
+           })
> vpts+2
[1] (-0.63, -0.82; 29) ( 0.18,  0.49; 39)
[3] (-0.84,  0.74; 59) ( 1.60,  0.58; 93)
[5] ( 0.33, -0.31; 22)

```

Method Subset

Definisi method untuk metode subset:

```
> setMethod("[",
+           signature(x = "vcoords",
+                   i = "ANY",
+                   j = "missing",
+                   drop = "missing"),
+           function(x, i, j)
+             vcoords(xcoords(x)[i],
+                   ycoords(x)[i],
+                   nilai(x)[i]))
> vpts[1:3]
[1] (-0.63, -0.82; 27)
[2] ( 0.18,  0.49; 37)
[3] (-0.84,  0.74; 57)
```

Pemeriksaan Suatu Class Objek

- Untuk mengecek apakah suatu objek merupakan suatu class digunakan fungsi `is()`

```
> is(pts, "coords")           > is(vpts, "coords")
[1] TRUE                       [1] TRUE
> is(pts, "vcoords")         > is(vpts, "vcoords")
[1] FALSE                      [1] TRUE
```

- Untuk men-coerce objek ke objek lain dari suatu class digunakan fungsi `as()`

```
> as(vpts, "coords")
[1] (-0.63, -0.82) ( 0.18,  0.49) (-0.84,  0.74)
[4] ( 1.60,  0.58) ( 0.33, -0.31)
> as(pts, "vcoords")
[1] (-0.63, -0.82; ) ( 0.18,  0.49; )
[3] (-0.84,  0.74; ) ( 1.60,  0.58; )
[5] ( 0.33, -0.31; )
```

Akhir materi 3...