MATERI 5b
Pemrograman Berorientasi Objek (S4)

STK571 KOMPUTASI STATISTIK

Sistem Objek S4

- Mengatasi masalah dalam sistem objek 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"))
[1] "coords"
```

Menciptakan objek

- Dapat menggunakan fungsi new

```
> pts = new("coords",
            x = rnorm(5), y = rnorm(5))
```

- Tidak disarankan → lebih baik membuat suatu konstruktor

```
coords =
    function(x, y) {
        if (length(x) != length(y))
            stop("equal length x and y required")
        if (!is.numeric(x) || !is.numeric(y))
            stop("numeric x and y required")
        new("coords", x = as.vector(x),
                      y = as.vector(y))
    }
```

Ilustrasi

- Misal menciptakan objek pts

```
> pts = coords(round(rnorm(5), 2),
               round(rnorm(5), 2))
```

- Metode print → metode show

```
> pts
An object of class "coords"
Slot "x":
[1]  1.31  1.15 -1.15 -0.46 -0.97

Slot "y":
[1]  0.64 -0.31  0.87  0.51 -1.34
```

Akses terhadap slot

- Menggunakan operator @

```
> pts@x
[1]  1.31  1.15 -1.15 -0.46 -0.97

> pts@y
[1]  0.64 -0.31  0.87  0.51 -1.34
```

- Tidak disarankan secara langsung → Create fungsi aksesor

```
> xcoords = function(obj) obj@x
> ycoords = function(obj) obj@y
```

Fungsi Generik show

- Fungsi generik show setara dengan fungsi generik print pada sistem objek S3
- Penciptaan fungsi generik menggunakan fungsi setMethod
- Argumen didefinisikan dalam signature

```
> setMethod(show, signature(object = "coords"),
            function(object)
            print(data.frame(x = xcoords(object),
                             y = ycoords(object))))
[1] "show"
attr(,"package")
[1] "methods"
```

```
> pts
      x     y
1  1.31  0.64
2  1.15 -0.31
3 -1.15  0.87
4 -0.46  0.51
5 -0.97 -1.34
```

Definisi Fungsi Generik

- Menggunakan setGeneric

```
> setGeneric("display",
             function(obj)
             standardGeneric("display"))
[1] "display"
> setMethod("display", signature(obj = "coords"),
            function(obj)
            print(paste("(",
                        format(xcoords(obj)),
                        ", ",
                        format(ycoords(obj)),
                        ")", sep = ""),
                  quote = FALSE))
[1] "display"
```

```
> display(pts)
[1] ( 1.31,  0.64) ( 1.15, -0.31)
[3] (-1.15,  0.87) (-0.46,  0.51)
[5] (-0.97, -1.34)
```

Bounding Box

- Ilustrasi metode untuk mendapatkan batas dari coords

```
> 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:"))))
[1] "bbox"
```

```
> pts
      x     y
1  1.31  0.64
2  1.15 -0.31
3 -1.15  0.87
4 -0.46  0.51
5 -0.97 -1.34

> bbox(pts)
        x:    y:
min -1.15 -1.34
max  1.31  0.87
```

Inheritance

- Terdapat class baru yang diturunkan dari coords dengan menambahkan slot nilai

```
> setClass("vcoords",
           representation(value = "numeric"),
           contains = "coords")
[1] "vcoords"
```

Fungsi konstruktor

- Fungsi konstruktor dan aksesor

```
> vcoords =
      function(x, y, value)
      {
          if (!is.numeric(x) ||
              !is.numeric(y) ||
              !is.numeric(value) ||
              length(x) != length(value) ||
              length(y) != length(value))
                  stop("invalid arguments")
          new("vcoords", x = x, y = y,
              value = value)
      }

> values = function(obj) obj@value
```

Ilustrasi

- Instanisasi class vcoords:

```
> vpts = vcoords(xcoords(pts), ycoords(pts),
                 round(100 * runif(5)))
```

- Print/show objek masih dari class coords

```
> vpts
      x     y
1  1.31  0.64
2  1.15 -0.31
3 -1.15  0.87
4 -0.46  0.51
5 -0.97 -1.34
```

Polymorphism dari metode show

- Metode show baru untuk vcoords:

```
> setMethod(show, signature(object = "vcoords"),
            function(object)
            print(data.frame(
                    x = xcoords(object),
                    y = ycoords(object),
                    value = values(object))))
[1] "show"
attr(,"package")
[1] "methods"
```

```
> vpts
      x     y value
1  1.31  0.64     7
2  1.15 -0.31    79
3 -1.15  0.87    19
4 -0.46  0.51    98
5 -0.97 -1.34    79
```

Fungsi Matematika

- Menciptakan fungsi cos untuk nilai

```
> setMethod("cos", signature(x = "vcoords"),
            function(x)
            vcoords(xcoords(x),
                    ycoords(x),
                    cos(values(x))))
[1] "cos"
> cos(vpts)
      x     y      value
1  1.31  0.64  0.7539023
2  1.15 -0.31 -0.8959709
3 -1.15  0.87  0.9887046
4 -0.46  0.51 -0.8192882
5 -0.97 -1.34 -0.8959709
```

Fungsi Matematika

- Atau menggunakan group

```
> setMethod("Math", signature(x = "vcoords"),
            function(x)
            vcoords(xcoords(x),
                    ycoords(x),
                    callGeneric(values(x))))
[1] "Math"
```

```
> sqrt(vpts)
      x     y    value
1  1.31  0.64 2.645751
2  1.15 -0.31 8.888194
3 -1.15  0.87 4.358899
4 -0.46  0.51 9.899495
5 -0.97 -1.34 8.888194

> tan(vpts)
      x     y     value
1  1.31  0.64 0.8714480
2  1.15 -0.31 0.4956775
3 -1.15  0.87 0.1515895
4 -0.46  0.51 0.6998537
5 -0.97 -1.34 0.4956775
```

Fungsi dalam group Math

- Fungsi-fungsi yang termasuk dalam group Math

```
abs, sign, exp, sqrt, log, log10, log2,
cos, sin, tan, acos, asin, atan,
cosh, sinh, tanh, acosh, asinh, atanh,
ceiling, floor, trunc,
gamma, lgamma, digamma, trigamma
cumprod, cumsum, cummin, cummin.
```

Operasi Dua Buah Objek

- Terdapat dua group: Arith dan Compare
  - The arithmetic operators:

    `+, -, *, ^, %%, %/%, /`
  - The comparison operators:

    `==, >, <, !=, <=, >=`
- Keduanya berasal dari group yg lebih besar: Ops

Operasi Dua Buah Objek

- Operasi dilakukan untuk objek dari lokasi yang sama

```
> sameloc =
      function(e1, e2)
      (length(values(e1)) == length(values(e2))
      || any(xcoords(e1) == xcoords(e2))
      || any(ycoords(e1) == ycoords(e2)))
```

Operasi Dua Buah Objek

```
> setMethod("Arith", signature(e1 = "vcoords",
                               e2 = "vcoords"),
      function(e1, e2)
      {
          if (!sameloc(e1, e2))
              stop("identical locations required")
          vcoords(xcoords(e1),
                  ycoords(e1),
                  callGeneric(values(e1),
                              values(e2)))
      })
[1] "Arith"
```

```
> vpts
      x     y value
1  1.31  0.64     7
2  1.15 -0.31    79
3 -1.15  0.87    19
4 -0.46  0.51    98
5 -0.97 -1.34    79

> vpts + vpts
      x     y value
1  1.31  0.64    14
2  1.15 -0.31   158
3 -1.15  0.87    38
4 -0.46  0.51   196
5 -0.97 -1.34   158
```

```
> setMethod("Compare", signature(e1 = "vcoords",
                                 e2 = "vcoords"),
      function(e1, e2)
      {
          if (!sameloc(e1, e2))
              stop("identical locations required")
          callGeneric(values(e1), values(e2))
      })
[1] "Compare"
```

Penambahan Metode Arith

```
> setMethod("Arith",
      signature(e1 = "numeric",
                e2 = "vcoords"),
     function(e1, e2) {
         if (length(e1) > length(values(e2)))
             stop("incompatible lengths")
         vcoords(xcoords(e2),
                 ycoords(e2),
                 callGeneric(as.vector(e1),
                             values(e2)))
             })
[1] "Arith"
```

Memeriksa class

- Menggunakan fungsi is

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

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

Coercion objek

- Menggunakan fungsi as

```
> as(vpts, "coords")
      x     y
1  1.31  0.64
2  1.15 -0.31
3 -1.15  0.87
4 -0.46  0.51
5 -0.97 -1.34
```

Subset

- Mendefinisikan Operator Subset "["

```
> setMethod("[",
            signature(x = "vcoords",
                      i = "ANY",
                      j = "missing",
                      drop = "missing"),
            function(x, i, j)
            vcoords(xcoords(x)[i],
                    ycoords(x)[i],
                    values(x)[i]))
[1] "["
```

```
> vpts[1:3]
      x     y value
1  1.31  0.64     7
2  1.15 -0.31    79
3 -1.15  0.87    19

> vpts[values(vpts) > 50]
      x     y value
1  1.15 -0.31    79
2 -0.46  0.51    98
3 -0.97 -1.34    79
```

AKHIR MATERI