

Pengenalan Bahasa Pemrograman Julia

Fadjar Fathurrahman

Daftar Isi

1	Sistem Tipe	5
1.1	Deklarasi tipe	5

Bab 1

Sistem Tipe

Sistem tipe pada Julia bersifat dinamik, namun juga dapat mengambil keuntungan dari sistem tipe statik karena juga memungkinkan kita untuk mengindikasikan tipe dari suatu variabel atau data.

Secara default, tipe data pada Julia dapat dihilangkan sehingga suatu variabel dapat memiliki tipe apa saja.

1.1 Deklarasi tipe

Deklarasi tipe pada Julia dapat dilakukan dengan menggunakan operator `::`. Misalkan:

```
α::Float64
Ω::Complex64
N::Int64 = 10
a::Float64 = 34.1
```

Ketika operator `::` diberikan setelah suatu ekspresi, maka operator `::` dibaca sebagai *is an instance of* yang digunakan sebagai penegasan suatu ekspresi akan memiliki suatu tipe tertentu.

Deklarasi juga dapat disematkan pada definisi fungsi:

```
function sinc(x)::Float64
    if x == 0
        return 1
    end
    return sin(pi*x) / (pi*x)
end
```

Tipe abstrak

Tipe abstrak tidak dapat diinstansiasi dan hanya berfungsi sebagai node pada graf tipe, sehingga mendeskripsikan himpunan dari tipe-tipe konkrit yang saling berkaitan: tipe konkrit yang merupakan turunan dari tipe abstrak ini. Meskipun tidak memiliki instansiasi, tipe abstrak diperlukan karena tipe abstrak merupakan tulang punggung dari sistem tipe, sebagai hierarki konseptual yang membuat tipe sistem pada Julia lebih dari sekedar kumpulan dari implementasi obyek.

Sintaks:

```
abstract type NamaTipe end
abstract type NamaTipe <: TipeSuper end
```

Tipe primitif

Tipe primitif adalah tipe konkrit yang memiliki data plain old bits.

Tipe komposit

Biasa disebut dengan record, struct, atau object pada bahasa pemrograman yang lain.

```
struct MyCompositeType
    Ndata::Int64
    v::Array{Int64,2}
end
```

Fungsi `fieldnames()` dapat digunakan untuk mengetahui field apa saja yang ada pada suatu instans dari tipe komposit.

Objek komposit yang dideklarasikan dengan `struct` bersifat **immutable**: artinya objek tersebut tidak dapat dimodifikasi setelah konstruksi.

Misc

Untuk mengkonversi dari suatu tipe ke tipe yang lain dapat digunakan fungsi `convert()`.

Fungsi `typeof()` dapat digunakan untuk mengetahui tipe suatu variabel:

```
julia> typeof(3.14)
Float64
```

```
julia> typeof(11)
Int64
```

```
julia> typeof(2//5)
Rational{Int64}
```

```
julia> typeof( (1,4,2) )
Tuple{Int64,Int64,Int64}
```

Fungsi `“subtypes()”` dapat digunakan untuk mengetahui subtype dari suatu tipe dan `“supertype()”` digunakan untuk mengetahui supertipe dari suatu tipe.