KOM204 : BAHASA PEMROGRAMAN

Pemrograman Fungsional

Pertemuan 2

Learning Outcomes

- Mahasiswa dapat menuliskan fungsi dengan notasi Lambda Calculus
- Mahasiswa dapat menuliskan ekspresi dengan notasi Cambridge Prefix
- Mahasiswa dapat menggunakan interpreter Scheme untuk menulis program fungsional

Outline Materi

Pustaka acuan:

- Tucker & Noonan, Chapter 14

Outline Materi:

- Prinsip dasar pemrograman fungsional
- Lambda calculus
- Pemrograman Scheme
- Ekspresi Cambridge prefix

Functional Programming

- Program serba fungsi, artinya setiap persoalan diselesaikan dengan menggunakan fungsi.
- Mulai dikembangkan tahun 1960an, dimotivasi oleh peneliti bidang artificial intelligence, symbolic computation, theorem proving, rule-based system, dan NLP.
- Bahasa fungsional pertama adalah Lisp (McCarthy, 1960).
- Memodelkan masalah komputasi sebagai suatu fungsi matematika, yang mempunyai input (domain) dan hasil atau output (range).

Function & Lambda Calculus

- Contoh fungsi matematika:

  Kuadrat(n) = n * n

- *Kuadrat* adalah fungsi yang memetakan suatu himpunan bilangan real **R** (domain) ke bilangan real **R** (range). Atau secara formal didefinisikan:

  Kuadrat: **R** → **R**

- Suatu fungsi disebut total jika terdefinisi untuk semua elemen dalam domain, dan disebut parsial untuk selainnya.
- Fungsi *Kuadrat* adalah total karena domainnya seluruh himpunan bilangan real.

Function & Lambda Calculus

- Dasar dari pemrograman fungsional adalah lambda calculus.
- Ekspresi lambda menunjukkan parameter dan definisi dari fungsi, BUKAN nama fungsi.
- Contoh, ekspresi lambda yang mendefinisikan fungsi Kuadrat adalah:

  $(\lambda x . x * x)$

- Identifier x adalah parameter yang digunakan dalam fungsi (tanpa nama) x * x.
- Contoh aplikasinya:

  $((\lambda x . x * x)2)$

  menjadi 4.

Lambda Calculus

- Setiap identifier adalah ekspresi lambda.
- Jika M dan N adalah ekspresi lambda, maka aplikasi dari M dan N, ditulis sebagai (M N), adalah ekspresi lambda.
- Abstraksi, ditulis sebagai (λx . M), dimana x adalah identifier dan M adalah ekspresi lambda, juga merupakan ekspresi lambda.
- BNF grammar:

LambdaExpression → *variable* | (M N) | (λ *variable* . M)

M → *LambdaExpression*

N → *LambdaExpression*

Lambda Calculus

- Contoh ekspresi lambda:

  x

  (λx . x)

  ((λx . x) (λy . y))

- Aplikasi fungsi:

  ((λx . * x x)5) = (* 5 5)

Scheme

- Lisp (List Programming) adalah bahasa pemrograman fungsional yang asli. Saat ini mempunyai dua varian, yaitu:
  - ✓ Common Lisp (Steele, 1990)
  - ✓ Scheme (Kelsey, 1998) – http://www.drscheme.org/
- Tidak memiliki pernyataan assignment.
- Program ditulis sebagai fungsi rekursif.

Scheme

Ekspresi

- Ekspresi disusun dalam notasi Cambridge-prefix.
- Contoh:

  (+ 2 2)

- Operasi aritmatika:

(+)	0
(+ 5)	5
(+ 5 4 3 2 1)	15
(*)	1
(* 5)	5
(* 1 2 3 4 5)	120

- Contoh lain: (+ (* 5 4) (- 6 2)

Scheme

Variabel global

- Didefinisikan dengan menggunakan fungsi *define*.
- Contoh:

```
(define f 120)
```

Evaluasi ekspresi

f	120
(+ f 5)	125
(f)	error, kenapa?
5	5
#f	false
#t	true

Scheme

Evaluasi ekspresi

```
(define warna (quote (merah kuning hijau)))
(define warna '(merah kuning hijau))

(define x f)                    120
(define x ' f)                  x berisi simbol f
(define warna ' merah)
(define warna merah)            error, kenapa?
```