

KOM204 : BAHASA PEMROGRAMAN**Struktur Program Fungsional**

Pertemuan 3

Julio Adisantoso, ILKOM-IPB

Learning Outcomes

- Mahasiswa dapat menulis program fungsional menggunakan Scheme
- Mahasiswa dapat memahami dan menggunakan tipe data dalam pemrograman fungsional
- Mahasiswa dapat memahami dan menggunakan struktur kondisional dalam pemrograman fungsional

Julio Adisantoso, ILKOM-IPB

Outline Materi

Pustaka acuan:

- Tucker & Noonan, Chapter 14
- Cristian Giumale

Outline Materi:

- PLT Scheme
- Define function
- Conditional
- Tipe data
- Latihan

Julio Adisantoso, ILKOM-IPB

PLT Scheme

- PLT Scheme memiliki dua tools utama
 - ▶ MzScheme : the core compiler, interpreter, and run-time system
 - ▶ DrScheme : the programming environment
- DrScheme memiliki beberapa variant. Untuk menggunakan Scheme standar:
 - ▶ Pilih Module (Choose Language | Module)
 - ▶ Definisikan #lang scheme pada definition area.

Julio Adisantoso, ILKOM-IPB

Define

- Bentuk:

```
( define <id> <expr> )
( define ( <id> <id>* ) <expr>* )
```

- Contoh:

```
#lang scheme
(define x 5)           ; x sbg var bernilai 5
(define (potong st)   ; potong sbg fungsi
  (substring st 0 x))

> x
5
> (potong "hello world")
"hello"
```

Julio Adisantoso, ILKOM-IPB

Conditionals : if, and, or, cond

- Nilai Boolean:

```
▶ #t   true
▶ #f   false
```

- Bentuk IF

```
( if <expr> <expr> <expr> )
```

- Contoh:

```
(define (maks a b)
  (if (> a b) a b))

> (maks 5 10)
10
```

Julio Adisantoso, ILKOM-IPB

Conditionals : if, and, or, cond

- Bentuk AND dan OR:


```
( and <expr>* )
( or <expr>* )
```
- Contoh:


```
(define (test n)
  (if (and (>= n 5) (<= n 20))
      "di dalam" "di luar"
  )
)
> (test -4)
"di luar"
```

Julio Adisantoso, ILKOM-IPB

Conditionals : if, and, or, cond

- Bentuk cond:


```
( cond {[ <expr> <expr>* ]}* )
```
- Contoh:


```
(define (mutu n)
  (cond
    ((= n 4) "A")
    ((= n 3) "B")
    ((= n 2) "C")
    ((= n 1) "D")
    ((= n 0) "E")
    (#t "BL")
  )
)
```

Julio Adisantoso, ILKOM-IPB

Conditionals : when, unless

- Bentuk:


```
( when <test-expr> <then-expr> )
( unless <test-expr> <then-expr> )
```
- Contoh:


```
(define (bagi a b)
  (when (not (= b 0))
    (/ a b)
  )
)
; atau
(define (bagi a b)
  (unless (= b 0)
    (/ a b)
  )
)
```

Julio Adisantoso, ILKOM-IPB

Type Data

- Booleans
- Numbers
- Characters
- Strings (Unicode)
- Bytes and Byte Strings
- Symbols
- Keywords
- Pairs and Lists
- Vectors
- Hash Tables
- Boxes
- Void and Undefined

Julio Adisantoso, ILKOM-IPB

Type Data :: Boolean

- Nilai output : #t dan #f
- Fungsi `boolean?` Untuk memeriksa apakah suatu nilai yang diperiksa adalah boolean.
- Contoh


```
> (= 2 (+ 1 1))
#t
> (boolean? #t)
#t
> (boolean? #f)
#t
> (boolean? "no")
#f
> (if "no" 1 0)
1
```

Julio Adisantoso, ILKOM-IPB

Type Data :: Numbers

- Bilangan dalam Scheme:
 - ▶ Exact
 - Integer : 5, 9999999999999999, -17
 - Ratio : 1/2, 9999999999999999/2, -3/4
 - Complex number : 1+2i, 1/2+3/4i
 - ▶ Inexact : 2.0, 3.14e+87
- Fungsi untuk memeriksa :


```
integer?, rational?, real?, complex?,
number?
```
- Fungsi tipe data:


```
#e (exact), #i (inexact), #b (binary), #o (octal), #x
(hexa)
```

Julio Adisantoso, ILKOM-IPB

Tipe Data :: Numbers

```
> 0.5
0.5
> #e0.5
1/2
> #x03BB
955
> (/ 1 2)
1/2
> (/ 1 2.0)
0.5
> (if (= 3.0 2.999) 1 2)
2
> (inexact->exact 0.1)
3602879701896397/36028797018963968
```

Julio Adisantoso, ILKOM-IPB

Tipe Data :: Character

- Character dalam Scheme = Unicode scalar value.
- Jenis
 - ▶ Printable char : #\ diikuti dgn char
 - ▶ Unprintable char : #\u diikuti dgn hexa
- Beberapa Fungsi:
 - ▶ char->integer
 - ▶ integer->char
 - ▶ display
 - ▶ char-alphabetic?, char-numeric?, char-whitespace?
 - ▶ char-downcase, char-upcase
 - ▶ char=? Bandingkan dua atau lebih char
 - ▶ char-ci=? Tidak memperhatikan upper/lower case
 - ▶ eqv? Sama dengan equal? untuk char

Julio Adisantoso, ILKOM-IPB

Tipe Data :: Character

```
> (integer->char 65)
#\A
> (char->integer #\A)
65
> #\u03BB
#\λ
> (integer->char 17)
#\u0011
> (char->integer #\space)
32
> #\A
#\A
> (display #\A)
A
```

Julio Adisantoso, ILKOM-IPB

Tipe Data :: Character

```
> (char-alphabetic? #\A)
#t
> (char-numeric? #\0)
#t
> (char-whitespace? #\newline)
#t
> (char-downcase #\A)
#\a
> (char=? #\a #\A)
#f
> (char-ci=? #\a #\A)
#t
> (eqv? #\a #\A)
#f
```

Julio Adisantoso, ILKOM-IPB

Tipe Data :: String

- Array of characters.
- Contoh:


```
> "Apple"
"Apple"
> "\u03BB"
"λ"
> (display "Apple")
Apple
> (display "a \"quoted\" thing")
a "quoted" thing
> (display "two\nlines")
Two
lines
> (display "\u03BB")
λ
```

Julio Adisantoso, ILKOM-IPB

Tipe Data :: String

- Contoh:


```
> ;ambil char pada posisi 0
> (string-ref "Apple" 0)
#\A
> ;membuat string
> (define s (make-string 5 #\.))
> s
"....."
> ;mengubah character pada string
> (string-set! s 2 #\A)
> s
"..A.."
> (string<? "apple" "Banana")
#f
> (string-ci<? "apple" "Banana")
#t
> (string-upcase "Strasse")
"STRASSE"
```

Julio Adisantoso, ILKOM-IPB

Fungsi Lambda

- Bentuk:


```
(lambda (arg ...)
  body ...+)
```
- Contoh:


```
> ((lambda (a b) (+ a b)) 2 7)
9
```

Julio Adisantoso, ILKOM-IPB

Local Binding

- Bentuk:


```
(let ((var val) ...) e1 e2 ...)
```
- Contoh:


```
> (let ((x 5)) (* x 2))
10
(let ((x 10) (z 5)) (* x z))
50
```

Julio Adisantoso, ILKOM-IPB

Grouping Statements | Sequence

- Bentuk:


```
(begin e1 e2 ... en)
```
- Contoh:


```
(define (test a b)
  (begin
    (display a) (display "+")
    (display b) (display "=")
    (display (+ a b)) (newline)
  )
)
> (test 5 8)
5+8=13
```

Julio Adisantoso, ILKOM-IPB

Latihan (1)

Buat program membaca nama dan menampilkan seperti contoh.

```
> (halo "Scheme Nurachem")
Selamat Datang Scheme Nurachem
KOM204
```

```
#lang scheme
(define (halo st)
  (printf "Selamat Datang ~a\nKOM204" st )
)
```

Julio Adisantoso, ILKOM-IPB

Latihan (2)

Buat program menjumlahkan tiga bilangan.

```
> (jumlah 4 6 8)
18
```

```
#lang scheme
(define (jumlah a b c)
  (+ a b c)
)
```

Julio Adisantoso, ILKOM-IPB

Latihan (3)

Buat program menentukan bilangan terkecil dari tiga bilangan.

```
> (min 6 4 6)
4
```

```
#lang scheme
(define (min a b c)
  (if (and (< a b) (< a c))
      a
      (if (< b c) b c)
  )
)
```

Julio Adisantoso, ILKOM-IPB

Latihan (4)

Buat program menentukan salah satu akar persamaan kuadrat $ax^2+bx+c=0$.

```
> (akar 1 1 -6)
2
```

```
#lang scheme
(define (det a b c)
  (sqrt (- (* b b) (* 4 a c))))
(define (akar a b c)
  (/ (+ (* -1 b) (det a b c)) (* 2 a)))
```

Julio Adisantoso, ILKOM-IPB

Latihan (5)

Buat program menghitung nilai faktorial

```
> (faktorial 4)
24
```

```
#lang scheme
(define (faktorial n)
  (if (= n 0)
      1
      (* n (faktorial (- n 1)))))
```

Julio Adisantoso, ILKOM-IPB

Latihan (6)

Buat program menentukan nilai fibonacci ke-n

```
> (fib 10)
55
```

```
#lang scheme
(define (fib n)
  (cond
    ((= n 0) 0)
    ((= n 1) 1)
    (#t (+ (fib (- n 2)) (fib (- n 1)))))
  )
```

Julio Adisantoso, ILKOM-IPB