



Type Bentukan



Tujuan

- Mahasiswa mampu memahami definisi type bentukan
- Berdasarkan definisi yang dipahami, mahasiswa mampu membuat program yang memanfaatkan type bentukan
- Mahasiswa mampu mengimplementasikan type bentukan dalam LISP → melalui praktikum



Type

- Type adalah **himpunan nilai** dan sekumpulan **operator** yang terdefinisi terhadap type tersebut
 - Dalam konteks fungsional: operator dijabarkan dalam bentuk **fungsi**
- Jenis-jenis type:
 - Type dasar → sudah tersedia: integer, real, character, boolean
 - Type bentukan → dibuat sendiri



Type Bentukan

- **Himpunan nilai**: ditentukan oleh domain nilai komponennya → bisa type dasar, bisa type bentukan
- Menyatakan nilai type bentukan: dalam bentuk **tuple**.
 - Contoh: nilai suatu Point didefinisikan oleh tuple $\langle x:\text{integer}, y:\text{integer} \rangle$, misal: $\langle 0,0 \rangle$
- **Operator**: harus dibuat definisi, spesifikasi, dan realisasinya sendiri tergantung pada operasi yang berlaku pada type bentukan

Type Bentukan



- Beberapa jenis type bentukan:
 - Type yang memberikan nama baru type lain/
membatasi domain type lain
 - Contoh: type bilbulat : integer → memberi nama baru
 - type IntPos : integer > 0 → membatasi domain
 - Type yang terdiri atas komponen-komponen bertype lain
 - Contoh: type Point : <x : integer, y : integer>
 - Type bentukan tanpa nama
 - Contoh: range dari fungsi HHMMDD (diktat hlm. 44)
HHMMDD: integer[0..99999] → <integer \geq 0, integer[0..23],
2 integer[0..59]>



Mendefinisikan Type

- Dalam konteks fungsional mendefinisikan type adalah mendefinisikan:
 - Nama dan struktur type (komponen-komponennya)
 - Selektor untuk mengakses komponen-komponen type
 - Konstruktor untuk “membentuk” type
 - Predikat untuk menentukan karakteristik dan pemeriksaan besaran
 - Fungsi-fungsi lain yang didefinisikan untuk type tersebut



Type Point

- Diktat hlm. 37-38
 - Definisi type Point
 - Definisi dan spesifikasi Selektor
 - Definisi dan spesifikasi Konstruktor
 - Definisi, spesifikasi, realisasi Predikat
 - Contoh: IsOrigin?
 - Definisi, spesifikasi, realisasi Fungsi Lain
 - Contoh: Jarak; Jarak0; Kuadran

TYPE POINT



DEFINISI TYPE

type point: $\langle x: \text{real}, y: \text{real} \rangle$

$\{ \langle x, y \rangle \text{ adalah sebuah point, dengan } x \text{ adalah absis, } y \text{ adalah ordinat} \}$

DEFINISI DAN SPESIFIKASI SELEKTOR

Absis: $\text{point} \rightarrow \text{real}$

$\{ \text{Absis}(P) \text{ memberikan absis Point } P \}$

Ordinat: $\text{point} \rightarrow \text{real}$

$\{ \text{Ordinat}(P) \text{ memberikan ordinat Point } P \}$

DEFINISI DAN SPESIFIKASI KONSTRUKTOR

MakePoint: $2 \text{ real} \rightarrow \text{point}$

$\{ \text{MakePoint}(a, b) \text{ membentuk sebuah point dari } a \text{ dan } b \text{ dengan } a \text{ sebagai absis dan } b \text{ sebagai ordinat} \}$

DEFINISI DAN SPESIFIKASI PREDIKAT

IsOrigin?: $\text{point} \rightarrow \text{boolean}$

$\{ \text{IsOrigin?}(P) \text{ benar jika } P \text{ adalah titik origin yaitu titik } \langle 0, 0 \rangle \}$



DEFINISI OPERATOR/FUNGSI LAIN TERHADAP POINT

Jarak: 2 point \rightarrow real

{Jarak(P1,P2): menghitung jarak antara 2 point P1 dan P2}

Jarak0: point \rightarrow real

{Jarak0(P1) menghitung jarak titik terhadap titik pusat koordinat (0,0)}

Kuadran: point \rightarrow integer[1..4]

{Kuadran(P) menghitung mana kuadran dimana titik tersebut terletak.

Syarat P bukan titik origin dan bukan terletak pada sumbu X dan bukan terletak pada sumbu Y}

{Fungsi antara yang dipakai: FX2 adalah pangkat dua yang pernah didefinisikan pada least square dan SQRT(X) adalah fungsi dasar untuk menghitung akar}



REALISASI

IsOrigin?(x,y): $x=0$ and $y=0$

IsOrigin?(P): Absis(P)=0 and Ordinat(P)=0

Jarak 2 titik= $\sqrt{((x_1-x_2)^2 + (y_1-y_2)^2)}$

Jarak(P1,P2):

SQRT (FX2 (Absis(P1) – Absis(P2)) +
FX2 (Ordinat(P1) – Ordinat(P2))
)

Jarak dgn titik 0 = $\sqrt{(x^2 + y^2)}$

Jarak0(P):

SQRT (FX2 (Absis(P1)) + FX2 (Ordinat(P1)))

Kuadran(P):

depend on Absis(P), Ordinat(P):

Absis(P) > 0 and Ordinat(P) > 0: 1

Absis(P) < 0 and Ordinat(P) > 0: 2

Absis(P) < 0 and Ordinat(P) < 0: 3

Absis(P) > 0 and Ordinat(P) < 0: 4

Kuadran(x,y):

depend on x,y:

$x > 0$ and $y > 0$: 1

$x < 0$ and $y > 0$: 2

$x < 0$ and $y < 0$: 3

$x > 0$ and $y < 0$: 4



Type Pecahan

- Diktat hlm. 38-39
 - Definisi type Pecahan
 - Definisi dan spesifikasi Selektor
 - Definisi dan spesifikasi Konstruktor
 - Definisi, spesifikasi, realisasi Predikat
 - Contoh: $IsEqP?$; $IsLtP?$
 - Definisi, spesifikasi, realisasi Fungsi Lain
 - Contoh: $AddP$; $DivP$



TYPE PECAHAN

DEFINISI DAN SPESIFIKASI TYPE

type pecahan: $\langle n: \text{integer} \geq 0, d: \text{integer} > 0 \rangle$

{ $\langle n: \text{integer} \geq 0, d: \text{integer} > 0 \rangle$ n adalah pembilang (numerator) dan d adalah penyebut (denominator). Penyebut sebuah pecahan tidak boleh nol}

DEFINISI DAN SPESIFIKASI SELEKTOR

Pemb: pecahan $\rightarrow \text{integer} \geq 0$

{Pemb(p) memberikan numerator pembilang n dari pecahan tsb}

Peny: pecahan $\rightarrow \text{integer} > 0$

{Peny(p) memberikan denominator penyebut d dari pecahan tsb}

DEFINISI DAN SPESIFIKASI KONSTRUKTOR

MakeP: $\text{integer} \geq 0, \text{integer} > 0 \rightarrow \text{pecahan}$

{MakeP(x, y) membentuk sebuah pecahan dari pembilang x dan penyebut y dengan x dan y integer}



DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP PECAHAN

{Operator Aritmatika Pecahan}

AddP: 2 pecahan \rightarrow pecahan

*{AddP(P1,P2): menambahkan dua buah pecahan P1 dan P2:
 $n1/d1 + n2/d2 = (n1*d2 + n2*d1) / (d1*d2)}$ }*

SubP: 2 pecahan \rightarrow pecahan

*{SubP(P1,P2): mengurangi dua buah pecahan P1 dan P2:
 $n1/d1 - n2/d2 = (n1*d2 - n2*d1) / (d1*d2)}$ }*

MulP: 2 pecahan \rightarrow pecahan

*{MulP(P1,P2): mengalikan dua buah pecahan P1 dan P2:
 $(n1/d1) * (n2/d2) = (n1*n2) / (d1*d2)}$ }*

DivP: 2 pecahan \rightarrow pecahan

*{DivP(P1,P2): membagi dua buah pecahan P1 dan P2:
 $(n1/d1) / (n2/d2) = (n1*d2) / (d1*n2)}$ }*

RealP: pecahan \rightarrow real

{Menuliskan bilangan pecahan dalam notasi desimal}



DEFINISI DAN SPESIFIKASI PREDIKAT

{Operator relasional Pecahan}

IsEqP?: 2 pecahan \rightarrow boolean

{IsEqP?(P1,P2) true jika $P1=P2$

membandingkan 2 pecahan sama nilainya $n1/d1 = n2/d2$

*jika dan hanya jika $n1*d2 = n2*d1$ }*

IsLtP?: 2 pecahan \rightarrow boolean

{IsLtP?(P1,P2) true jika $P1 < P2$

membandingkan 2 pecahan, apakah $P1 < P2$: $n1/d1 < n2/d2$

*jika dan hanya jika $n1*d2 < n2*d1$ }*

IsGtP?: 2 pecahan \rightarrow boolean

{IsGtP?(P1,P2) true jika $P1 > P2$

membandingkan 2 pecahan, apakah $P1 > P2$: $n1/d1 > n2/d2$

*jika dan hanya jika $n1*d2 > n2*d1$ }*

$$\text{AddP}(n1,d1,n2,d2): n\text{-baru} = (n1*d2 + n2*d1)$$

$$d\text{-baru} = (d1*d2)$$



REALISASI

AddP(P1,P2): MakeP(

(Pemb(P1)*Peny(P2) + Pemb(P2)*Peny(P1)),

(Peny(P1)*Peny(P2))

Constructor untuk membuat Pecahan baru sebagai hasil pertambahan

Penyebut baru

Pembilang baru

SubP(P1,P2): MakeP(

(Pemb(P1)*Peny(P2) - Pemb(P2)*Peny(P1)),

(Peny(P1)*Peny(P2))

)

MulP(P1,P2): MakeP(

(Pemb(P1)*Pemb(P2)),

(Peny(P1)*Peny(P2))

)

DivP(P1,P2): MakeP(

(Pemb(P1)*Peny(P2)),

(Peny(P1)*Pemb(P2))

)



REALISASI

RealP(P): $\text{Pemb}(P)/\text{Peny}(P)$

IsEqP?(P1,P2):

$$\text{Pemb}(P1) * \text{Peny}(P2) = \text{Peny}(P1) * \text{Pemb}(P2)$$

IsLtP?(P1,P2):

$$\text{Pemb}(P1) * \text{Peny}(P2) < \text{Peny}(P1) * \text{Pemb}(P2)$$

IsGtP?(P1,P2):

$$\text{Pemb}(P1) * \text{Peny}(P2) > \text{Peny}(P1) * \text{Pemb}(P2)$$



Type Date

- Diktat hlm. 40-43
 - Definisi type Date
 - Penggunaan type bentukan lain
 - Definisi dan spesifikasi Selektor
 - Definisi dan spesifikasi Konstruktor
 - Definisi, spesifikasi, realisasi Predikat
 - Contoh: IsBefore? (versi dengan let dan tanpa let)
 - Definisi, spesifikasi, realisasi Fungsi Lain
 - Contoh: Nextday

TYPE DATE



DEFINISI DAN SPESIFIKASI TYPE

type Hr: integer [1..31]

{definisi ini hanyalah untuk “menamakan” type integer dgn nilai tertentu supaya mewakili hari, shg jika dipunyai suatu nilai integer, kita dapat memeriksa apakah nilai integer tersebut mewakili Hari yg absah}

type Bln: integer [1..12]

{definisi ini hanyalah untuk “menamakan” type integer dgn nilai tertentu supaya mewakili bulan}

type Thn: integer > 0

{definisi ini hanyalah untuk “menamakan” type integer dgn nilai tertentu supaya mewakili tahun}

type date <d: Hr, m: Bln, y: Thn>

{<d,m,y> adalah tanggal d bulan m tahun y}

DEFINISI DAN SPESIFIKASI SELEKTOR

Day: date → Hr

{Day(D) memberikan hari d dari D yang terdiri dari <d,m,y>}

Month: date → Bln

{Month(D) memberikan bulan m dari D yang terdiri dari <d,m,y>}

Year: date → Thn

{Year(D) memberikan tahun y dari D yang terdiri dari <d,m,y>}



DEFINISI DAN SPESIFIKASI KONSTRUKTOR

MakeDate: $\langle \text{Hr}, \text{Bln}, \text{Thn} \rangle \rightarrow \text{date}$

{MakeDate $\langle h, b, t \rangle$ membentuk tanggal pada hari, bulan, tahun tertentu}

DEFINISI DAN SPESIFIKASI OPERATOR TERHADAP DATE

NextDate: $\text{date} \rightarrow \text{date}$

{NextDate(D): menghitung date yg merupakan keesokan hari dari date D yg diberikan, misal Nextday($\langle 30, 4, 80 \rangle$) adalah $\langle 1, 5, 80 \rangle$ }

Yesterday: $\text{date} \rightarrow \text{date}$

{Yesterday(D): menghitung date yg merupakan 1 hari sebelum date D yg diberikan, misal Yesterday($\langle 1, 3, 80 \rangle$) adalah $\langle 29, 2, 80 \rangle$ }

NextNDay: $\text{date}, \underline{\text{integer}} \rightarrow \text{date}$

{NextNDay(D): menghitung date yg merupakan hari N dari date D yg diberikan, misal NextNday($\langle 30, 4, 80 \rangle, 3$) adalah $\langle 3, 5, 80 \rangle$ }

HariKe1900: $\text{date} \rightarrow \underline{\text{integer}}[0..366]$

{HariKe1900(D): menghitung jumlah hari terhadap 1 Januari pada tahun y dengan memperhitungkan apakah y adalah tahun kabisat atau bukan}



DEFINISI DAN SPESIFIKASI PREDIKAT

IsEqD?: 2 date \rightarrow boolean

{IsEqD?(D1,D2) true jika $D1=D2$ yaitu $d1=d2$, $m1=m2$ dan $y1=y2$ }

IsBefore?: 2 date \rightarrow boolean

{IsBefore?(D1,D2) true jika D1 adalah sebelum D2}

IsAfter?: 2 date \rightarrow boolean

{IsAfter?(D1,D2) true jika D1 adalah sesudah D2}

IsKabisat?: **Thn** \rightarrow boolean

*{IsKabisat?(a) true jika tahun **a** adalah tahun kabisat: habis dibagi 4 tetapi tidak habis dibagi 100 atau habis dibagi 400}*



REALISASI

{hanya sebagian, sisanya silakan dibuat sebagai latihan}

IsBefore?(D1,D2): periksa tahunnya, jika sama maka periksa bulannya, jika sama maka periksa tanggalnya

IsBefore?(D1,D2):

Let J1=Day(D1), M1=Month(D1), T1=Year(D1),
J2=Day(D2), M2=Month(D2), T2=Year(D2)

in

if T1 ≠ T2 then
T1 < T2

else

if M1 ≠ M2 then
M1 < M2

else

J1 < J2

{Versi lain (Tanpa Let)}

IsBefore?(D1,D2):

if Year(D1) ≠ Year(D2) then
Year(D1) < Year(D2)

else

if Month(D1) ≠ Month(D2) then
Month(D1) < Month(D2)

else

Day(D1) < Day(D2)



REALISASI

IsEqD?(D1,D2): (Year(D1) = Year(D2))
and
(HariKe1900(D1) = HariKe1900(D2))

IsBefore?(D1,D2): if (Year(D1) = Year(D2)) then
 HariKe1900(D1) < HariKe1900(D2)
else
 Year(D1) < Year(D2)

IsAfter?(D1,D2): if (Year(D1) = Year(D2)) then
 HariKe1900(D1) > HariKe1900(D2)
else
 Year(D1) > Year(D2)

IsKabisat?(a):
 ((a mod 4 = 0) and (a mod 100 ≠ 0))
or
 (a div 400 = 0)



Type Bentuk digunakan dalam Type Bentuk Lain

- Contoh: Type Garis
 - Suatu Garis terdiri atas 2 buah Point

```
type Point : < x : real , y : real >
```

{ <x,y> adalah sebuah point/titik, dengan x adalah absis dan y adalah ordinat }

```
type Garis : < PAw : Point , PAkh : Point >
```

{ <PAw, PAkh> adalah sebuah garis dengan PAw adalah titik awal garis dan PAkh adalah titik akhir garis }



Type Bentukan tanpa nama

- Fungsi dapat mengembalikan range type bentukan tanpa nama
- Contoh: EkuivalensiDetik (diktat hlm. 44)
 - Versi dengan let dan fungsi antara
 - Versi dengan kalkulasi tanpa let dan fungsi antara, koreksi diktat:

```
HHMMDD (x): < x div 86400,  
              (x mod 86400) div 3600,  
              ((x mod 86400) mod 3600) div 60,  
              ((x mod 86400) mod 3600) mod 60 >
```




Translasi ke LISP



Definisi type

- Dalam LISP, tuple nilai suatu type bentukan dinyatakan dalam bentuk **list**
 - Contoh: nilai Point $\langle -1, 3 \rangle$ dinyatakan sebagai: $(-1 \ 3)$
- Karena list adalah salah satu symbol dasar dalam LISP, tidak ada realisasi definisi type
- Definisi type:
 - Dinyatakan secara eksplisit dalam bentuk komentar

```
; Definisi Type
; type Point : < x : real, y : real >
;   <x,y> adalah sebuah point, dengan x adalah absis
;   dan y adalah ordinat}
```

Selektor



- Definisi dan spesifikasi selektor diberikan dalam bentuk komentar
- Selektor direalisasikan sebagai fungsi untuk mengakses komponen-komponen list

```
; Definisi dan spesifikasi selektor
; Absis : Point -> real
;   Absis(P) memberikan absis dari point P
; Ordinat : Point -> real
;   Ordinat(P) memberikan ordinat dari point P

; Realisasi Selektor
(defun Absis (P) (car P))
(defun Ordinat (P) (car (cdr P)))
```



Konstruktor

- Definisi dan spesifikasi diberikan dalam bentuk komentar
- Konstruktor direalisasikan sebagai fungsi untuk membentuk suatu list

```
; Definisi dan spesifikasi konstruktor
; MakePoint : 2 real -> Point
;   MakePoint(a,b) membentuk sebuah Point dari a
;   dan b dengan a sebagai absis dan b sebagai
;   ordinat

; Realisasi Konstruktor
(defun MakePoint (a b) (list a b))
```



Predikat

- Definisi dan spesifikasi tuliskan dalam bentuk komentar
- Realisasi: sebagai fungsi yang menghasilkan nilai boolean

```
; Definisi dan spesifikasi Predikat
; IsOrigin? : Point -> boolean
;   IsOrigin?(P) benar jika P adalah titik origin
;   yaitu <0,0>

; Realisasi Predikat IsOrigin?
(defun IsOrigin? (P)
  (and (= (Absis P) 0)
        (= (Ordinat P) 0)
  )
)
```



Fungsi Lain

- Definisi dan spesifikasi tuliskan dalam bentuk komentar

```
; Definisi dan spesifikasi Fungsi Lain
; Kuadran : Point -> integer[1..4]
;   Kuadran (P) menghitung kuadran di mana suatu
;   point terletak.
;   Prekondisi: P bukan di titik origin dan bukan
;   terletak pada sumbu x dan sumbu y

; Realisasi Fungsi Kuadran
(defun Kuadran (P)
  (cond ((and (> (Absis P) 0) (> (Ordinat P) 0)) 1)
        ((and (< (Absis P) 0) (> (Ordinat P) 0)) 2)
        ((and (> (Absis P) 0) (< (Ordinat P) 0)) 3)
        ((and (< (Absis P) 0) (< (Ordinat P) 0)) 4)
  )
)
```

Aplikasi Type Bentukan



- Suatu ekspresi dapat di-assign suatu nilai type bentukan dengan menggunakan **setq**

```
⇒(setq P1 (MakePoint -3 1))  
(-3 1)  
⇒(Absis P1)  
-3  
⇒(Ordinat P1)  
1  
⇒(IsOrigin? P1)  
NIL  
⇒(Kuadran P1)  
2
```



Type bentukan tanpa nama

- Fungsi yang menghasilkan range type bentukan tanpa nama direalisasikan sebagai fungsi yang menghasilkan list

```
(defun HHMMDD (x)
  (list (div x 86400)
        (div (rem x 86400) 3600)
        (div (rem (rem x 86400) 3600) 60)
        (rem (rem (rem x 86400) 3600) 60))
)
```




Type bentukan tanpa nama

- Contoh: Aplikasi fungsi HHMMDD dari program EkivalensiDetik (diktat hlm.44)

```
⇒ (HHMMDD 86400)
(1 0 0 0)
⇒ (HHMMDD 43200)
(0 12 0 0)
```



Latihan Soal



Latihan Soal 1

- Didefinisikan type bentukan Point seperti pada diktat hlm. 37-38
- Dalam notasi fungsional: Buatlah definisi, spesifikasi, realisasi dari predikat **Translasi** yang menerima masukan sebuah Point (misalnya P) dan 2 buah angka real (misalnya dx dan dy) dan menghasilkan Point hasil translasi P sejauh dx pada arah sumbu x dan dy pada arah sumbu y
- Contoh:
 - Diketahui $P1 = \langle 0, 1 \rangle$
 - Translasi ($P1, 2, -2$) menghasilkan $\langle 2, -1 \rangle$



Latihan Soal 2

- Diketahui definisi dan spesifikasi fungsi **KuadranPrePostTranslasi** sbb:

```
KuadranPrePostTranslasi : Point, 2 real ->  
                           <integer, integer>
```

```
{ KuadranPrePostTranslasi(P, dx, dy) menghasilkan  
tuple <K1, K2>. K1 adalah kuadran P, sedangkan K2  
adalah kuadran P setelah ditranslasi sejauh dx searah  
sumbu x dan dy searah sumbu y. }
```

- Buatlah realisasinya dalam notasi fungsional.
- Hint: Gunakan fungsi antara.



Latihan Soal 3

- Realisasi fungsi **Yesterday** dari type bentukan Date pada diktat hlm. 43 masih mengandung kesalahan.
- Buatlah realisasi fungsi Yesterday versi Anda sendiri.



Tugas di Rumah

- Translasikan latihan soal hari ini dalam notasi LISP
- Modul Pra-Praktikum Bagian Fungsional
 - F-01. ADT POINT
 - F-02. ADT PECAHAN
 - F-03. ADT DATE

Tugas di Rumah



- Untuk dipikirkan:
 - JAM adalah salah satu type bentukan yang digunakan dalam banyak kasus
 - Contoh:
 - 14:20 (jam 14 menit 20)
 - 14:20:30 (jam 14 menit 20 detik 30)
 - 2:20:30 PM (= 14:20:30)
 - Bagaimana membuat type JAM yang memiliki komponen jam, menit, detik?
 - Bagaimana batasan domain nilai tiap komponen JAM?
 - Operator apa saja yang mungkin berlaku terhadap JAM?