

# Dasar Pemrograman

Part - 2

# Outline Course

- TYPE BENTUKAN
- LIST
- SET
- LIST of LIST
- Tree
- Binary Tree
- Lambda Expression

# TYPE BENTUKAN

- Type : Himpunan nilai dan sekumpulan operator terdefinisi terhadap type tersebut.
- Contoh:
  - Type point : terdiri dari <absis, ordinat> bertype <integer, integer>
  - Type jam : <jam, menit, detik>
  - Type date : <tanggal, bulan, tahun>

# TABEL

- Kumpulan elemen yang diorganisasi secara kontinu, dan setiap elemen dapat diakses melalui indeksnya
- Tabel dipakai untuk menguraikan fungsi yang terdefinisi pada interval bilangan bulat  $[bi...bs]$  dan memetakan pada harga tertentu.
  - $bi \leq bs$ ,  $bi$  = batas bawah,  $bs$  = batas atas
  - banyaknya elemen  $= bs - bi + 1 \rightarrow$  ukuran table
  - Contoh :  $[31, 24, 31, 90] \rightarrow$  table berdimensi 4
  - Seleksi terhadap elemen table dituliskan dengan indeks  $T_i$  atau  $T[i]$

# LIST

- Sekumpulan elemen list yang bertipe sama.
- Representasi list:
  - Teks (list of kata)
  - Kata (list of huruf)
  - Sequential file (list of record)
- List kosong adalah list yang tidak mempunyai elemen dengan notasi [ ].
- Dua buah list sama jika elemen list sama urutan dan nilainya

# Definisi dan spesifikasi (1)

- Type list :  $[ ]$ , atau  $[e \circ \text{List}] \rightarrow$  konstruktor menambahkan elemen di awal
- Type list :  $[ ]$ , atau  $[e \bullet \text{List}] \rightarrow$  konstruktor menambahkan elemen di akhir
- **Konso : elemen, List  $\rightarrow$  List**
  - Konso(e,L) : menghasilkan sebuah list dari e dan L, dengan e sebagai elemen pertama
- **Kons $\bullet$  : List, elemen  $\rightarrow$  List**
  - Kons(L,e) : menghasilkan sebuah list dari L dan e, dengan e sebagai elemen terakhir

# Definisi dan spesifikasi (2)

- **FirstElmt: List tidak kosong  $\rightarrow$  elemen**
  - FirstElmt(L) : menghasilkan elemen pertama list L
- **Tail : List tidak kosong  $\rightarrow$  List**
  - Tail(L) : menghasilkan list tanpa elemen pertama list L, mungkin kosong
- **LastElmt : list tidak kosong  $\rightarrow$  List**
  - LastElmt(L) : menghasilkan elemen terakhir list L
- **Head : List Tidak Kosong  $\rightarrow$** 
  - Head(L) : menghasilkan list tanpa elemen terakhir list L, mungkin kosong
- **IsEmpty : List  $\rightarrow$  Boolean**
  - IsEmpty(L) benar jika list kosong

# Definisi dan spesifikasi (3)

- **IsOneElmt(L)** : benar jika list L hanya mempunyai satu elemen
- **IsEqual(L1,L2)** : benar jika semua elemen list L1 sama dengan L2 : sama urutan dan sama nilainya
- **NbElmt(L)** : menghasilkan banyaknya elemen list, nol jika kosong
- **ElmtkeN (N,L)** : mengirimkan elemen list yang ke N
- **Copy(L)** : menghasilkan list yang identik dengan list asal
- **Inverse(L)** : menghasilkan yang dibalik



## Definisi dan spesifikasi (4)

- **Konkat(L1,L2) :** menghasilkan konkatenasi 2 buah list, dengan list L2 “sesudah” list L1
- **IsMember(X,L) :** benar jika X adalah elemen list L
- **IsNbElmtN(N,L) :** benar jika banyaknya elemen list sama dengan N
- **IsInverse(L1,L2) :** Benar jika L2 adalah list dengan urutan elemen terbalik dibandingkan L1
- **IsXElmtkeN(N,X,L) :** benar jika X adalah elemen list yang ke N

# Ilustrasi

L	4	'a'	-2	'xyz'	0.5
	L[0]	L[1]	L[2]	L[3]	L[4]

List L mempunyai 5 elemen

First\_element = 4

Tail\_element = 'a', -2, 'xyz', 0.5

Head\_element = 4, 'a', -2, 'xyz'

Last\_element = 0.5

NB\_element = 5