



Praktikum Struktur Data

Modul PSDA

PSDA-12a. ADT LIST RECURSIVE DARI FUNGSIONAL

Buatlah ADT List of Integer sesuai dengan definisi dan spesifikasi di bawah ini dalam Bahasa C. Sebagai referensi dapat dilihat pada Diktat Fungsional dan Suplemen PSDA-12a.

TYPE LIST

DEFINISI DAN SPESIFIKASI TYPE

{Konstruktor menambahkan elemen di awal, notasi prefix }

type List of integer : [], atau [e o List]

{Konstruktor menambahkan elemen di akhir, notasi postfix }

type List of integer : [] atau [List • e]

{Definisi List : sesuai dengan definisi rekursif di atas }

DEFINISI DAN SPESIFIKASI KONSTRUKTOR

Konso : integer, List of integer tidak kosong \rightarrow List of integer

*{Konso(e,L): menghasilkan sebuah list dari e dan L, dengan e sebagai elemen pertama e :
e oL \rightarrow L'}*

Kons• : List of integer tidak kosong, integer \rightarrow List of integer

*{Kons(L,e): menghasilkan sebuah list dari L dan e, dengan e sebagai elemen terakhir list
: L •e \rightarrow L'}*

DEFINISI DAN SPESIFIKASI SELEKTOR O (nol)

FirstElmt: List of integer tidak kosong \rightarrow integer

{FirstElmt(L) Menghasilkan elemen pertama list L }

Tail : List of integer tidak kosong \rightarrow List of integer

{Tail(L) : Menghasilkan list tanpa elemen pertama list L }

LastElmt : List of integer tidak kosong \rightarrow integer

{LastElmt(L) : Menghasilkan elemen terakhir list L }

Head : List of integer tidak kosong \rightarrow List of integer

{Head(L) : Menghasilkan list tanpa elemen terakhir list L }

DEFINISI DAN SPESIFIKASI PREDIKAT DASAR (UNTUK BASIS ANALISA



Praktikum Struktur Data

Modul PSDA

REKURENSI

{Basis 0 }

IsEmpty : List of integer \rightarrow boolean

{IsEmpty(L) benar jika list kosong }

{Basis 1 }

IsOneElmt: List of integer \rightarrow boolean

{IsOneElmt (X,L) adalah benar jika teks hanya mempunyai satu elemen }

DEFINISI DAN SPESIFIKASI PREDIKAT KEABSAHAN

IsListInt: List \rightarrow boolean

{IsListInt(L) menghasilkan true jika L adalah list dengan elemen integer }

DEFINISI DAN SPESIFIKASI PREDIKAT RELASIONAL

IsEqual : 2 List of integer \rightarrow boolean

{ IsEqual (L1,L2) benar jika semua elemen list L1 sama dengan L2: sama urutan dan sama nilainya }

BEBERAPA DEFINISI DAN SPESIFIKASI FUNGSI LAIN

NbElmt : List of integer \rightarrow integer

{NbElmt(L) : Menghasilkan banyaknya elemen list, nol jika kosong }

ElmtkeN : integer ≥ 0 , List \rightarrow integer

{ElmtkeN (N, L) : Mengirimkan elemen list yang ke N, $N \leq \text{NbELmt}(L)$ dan $N \geq 0$ }

Copy : List of integer \rightarrow List of integer

{ Copy(L) : Menghasilkan list yang identik dengan list asal}

Inverse : List of integer \rightarrow List of integer

{ Inverse(L) : Menghasilkan list L yang dibalik, yaitu yang urutan elemennya adalah kebalikan dari list asal}

Konkat : 2 List of integer \rightarrow List of integer

{Konkat(L1,L2) : Menghasilkan konkatenasi 2 buah list, dengan list L2 "sesudah" list L1}

BEBERAPA DEFINISI DAN SPESIFIKASI PREDIKAT



Praktikum Struktur Data

Modul PSDA

IsMember: integer, List of integer \rightarrow boolean

{IsMember (X,L) adalah benar jika X adalah elemen list L }

IsFirst: integer, List of integer tidak kosong \rightarrow boolean

{IsFirst (X,L) adalah benar jika X adalah elemen pertama list L }

IsLast : integer, List of integer tidak kosong \rightarrow boolean

{IsLast (X,L) adalah benar jika X adalah elemen terakhir list L }

IsNbElmtN : integer ≥ 0 dan \leq NBElmt(L), List \rightarrow boolean

{IsNbElmtN (N, L) true jika banyaknya elemen list sama dengan N }

IsInverse : List of integer, List of integer \rightarrow boolean

{IsInverse(L1,L2) true jika L2 adalah list dengan urutan elemen terbalik dibandingkan L1}

IsXElmtkeN : integer ≥ 0 dan \leq NBElmt(L), integer, List of integer \rightarrow boolean

{IsXElmtkeN (N, X,L) adalah benar jika X adalah elemen list yang ke N }

BEBERAPA DEFINISI DAN SPESIFIKASI OPERASI LAIN TERKAIT INTEGER Maxlist

: List of integer tidak kosong \rightarrow integer

{ Maxlist(Li) : menghasilkan elemen Li yang bernilai maksimum }

Dimensi : List of integer \rightarrow integer > 0

{ Dimensi(Li) menghasilkan banyaknya elemen list integer }

List+ : 2 list of integer $\geq 0 \rightarrow$ list of integer ≥ 0

{ List+ (Li1,Li2): menjumlahkan setiap elemen list integer yang berdimensi sama, hasilnya berupa list integer berdimensi tsb. }

Insert : list of integer \rightarrow list of integer

{ Insert(Li) menghasilkan list integer yang terurut dengan metoda insertion sort }

Insert : integer, list of integer tidak kosong dan terurut membesar \rightarrow list of integer terurut membesar

{ Insert(x,Li) melakukan insert x ke Li menghasilkan list terurut membesar }

maxNb : list of integer \rightarrow <integer, integer>

{maxNb(Li) menghasilkan <nilai maksimum, kemunculan nilai maksimum> dari suatu list integer Li ; <m,n> =m adalah maks x dari n # occurrence m dalam list}