



UJIAN AKHIR SEMESTER (UAS)
INFORMATIKA FAKULTAS SAINS DAN MATEMATIKA UNIVERSITAS
DIPONEGORO SEMARANG
SEMESTER GASAL TAHUN 2022/2023

Mata Kuliah	:	Struktur Data (4 sks)
Kelas	:	A, B, C, dan D
Pengampu	:	Dr. Aris Puji Widodo, MT. Edy Suharto, ST., M.Kom.
Departemen/Program Studi	:	Informatika/S1 Informatika
Hari/Tanggal	:	Selasa/13 Desember 2022
Jam/Ruang	:	09.30 – 11.00 (90 Menit)
Sifat Ujian	:	Open Books (<i>Tidak diperbolehkan membuka Handphone/PC/Laptop</i>)

Perhatian, Sebelum mengerjakan BACALAH di bawah ini

1. Kerjakanlah sendiri dengan jujur, jika terjadi kecurangan diberikan nilai NOL.
 2. Selain nomor 1, kerjakan dengan menuliskan NOTASI ALGORITMIK, jika terdapat notasi lain diberikan nilai NOL.
 3. Kerjakan dengan menggunakan PENSIL, jika menggunakan yang lain diberikan nilai NOL.
 4. Semua primitif-primitif pada masing-masing ADT yang diberikan diperkuliahan tidak perlu direalisasikan, anda dapat langsung menggunakannya.
1. Diberikan skema searching pada potongan algoritma di bawah ini, skema ini mengandung kesalahan **FATAL**. Berikan analisis dan penjelasan letak kesalahan **FATAL** nya!

```
Procedure ListSearchFatal (input L : List, input X : InfoType)
{Sequential Search nilai X pada sebuah List Linier L}
{Kamus Lokal}
  P : address {address pada pencarian be urutan, di mana X
               {ditemukan, dan P = Nil jika tidak ketemu}
  Found : boolean {True jika X ditemukan, dan False jika tidak}
{Algoritma}
  {List linier L sudah terdefinisi}
  P ← First(L) {inisialisasi}
  while (Next(P) ≠ Nil) and (X ≠ Info(P)) do
    P ← Next(P) {maju}
  { Next(P) = Nil or X = Info(P) }
  Found ← (X = Info(P)) {nilainya boolean}
```

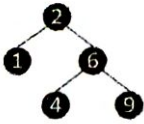
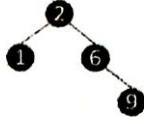
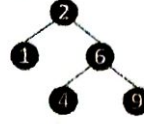
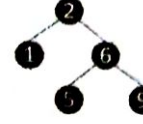
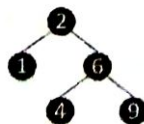
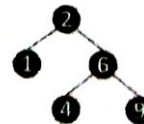
2. Diberikan sebuah **List Circular** yang memiliki type elemennya (InfoType) adalah bilangan integer.

Function IsListSimetri (L1: List, L2: List) → boolean
 {Mengirimkan **TRUE** jika setiap elemen pada L1 sama dengan L2 (**Simetri**)}
 {Ex. Jika **L1**: 1->2->3->5, **L2**: 1->2->3->6, maka **False**}
 { Jika **L1**: 1->2->3->5, **L2**: 1->2->3->5, maka **True**}
 {Kamus Lokal}
 {Algoritma}

3. Diberikan sebuah **List Double Pointer Linier** yang memiliki type elemennya (InfoType) adalah bilangan integer.

Procedure IntersectList (input: L1:List, input:L2:List, output:L3:List)
 {**IS**: List L1 dan L2 adalah List sembarang yang mungkin kosong}
 {**FS**: Menghasilkan List baru L3 yang merupakan **intersect** }
 { (adalah irisan 2 buah himpunan) dari elemen List L1 dan L2}
 {**Proses**: Buat List baru L3, cari setiap elemen L1 pada L2}
 { jika ketemu, lakukan insert ke L3}
 { Jika tidak ada yang ketemu menghasilkan List Kosong}
 {Ex. **L1**: 1->2->3->6, **L2**: 5->6->1->3->2, maka **L3**: 1->2->3->6}
 { **L1**: 1->2->3->6, **L2**: 5->8->11->13->12, maka **L3**: Nil}
 {Kamus Lokal}
 {Algoritma}

4. Diberikan sebuah **Pohon Biner** yang memiliki type elemennya (InfoType) adalah bilangan integer.

Function IsSimilarTree (P1: BinTree, P2: BinTree) → Boolean
 {mengirimkan **TRUE** jika P1 dan P2 sama struktur dan nilai elemennya}
 {P1 dan P2 adalah Pohon Biner sembarang yang mungkin kosong}
 IsSimilarTree(P1,P2)←False
 IsSimilarTree(P1,P2)←False

 Pohon Biner P1

 Pohon Biner P2

 Pohon Biner P1

 Pohon Biner P2
 IsSimilarTree(P1,P2)←True

 Pohon Biner P1

 Pohon Biner P2
 {
 {Kamus Lokal}
 {Algoritma}