

Tugas Pendahuluan Modul 5
STRUKTUR DATA - Ganjil 2025/2026
"Singly Linked List"

A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL**
4. / **TES ASESMEN**.
5. Deadline pengumpulan TP Modul 5 adalah Senin, 13 Oktober 2025 pukul 06.00 WIB.
6. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN.**
7. **DILARANG PLAGIAT (PLAGIAT = E).**
8. Kerjakan TP dengan jelas agar dapat dimengerti.
9. File diupload di LMS menggunakan format PDF dengan ketentuan:
TP_MOD_[XX]_NIM_NAMA.pdf

CP (WA):

- Falih (+62 857-2774-5199)
- Faried (+62 813-8921-4045)
- Arief (+62 852 1252 8394)
- Evan (+62 896-0298-0999)

SELAMAT MENGERJAKAN^^

B. Soal TP

1. Pembuatan ADT

Buatlah ADT Singly Linked List yang akan menyimpan data berupa angka bilangan bulat (integer).

```
type infotype : integer
type address : pointer to elmlist

type elmlist <
    info : infotype
    next : address
>

type List <
    first : address
>

procedure searchElement(in/out L : list, in x : infotype)
procedure selectionSort(in/out L : list)
procedure insertSorted(in/out L : list, in p : address)
```

2. Mencari Elemen Tertentu dalam List

Buatlah prosedur **searchElement** yang mengizinkan pengguna untuk mencari apakah sebuah nilai tertentu ada dalam list.

```
procedure searchElement(in/out L : list, in x : infotype)
{I.S. Terdefinisi List L yang tidak kosong.
 F.S. Menampilkan alamat dan posisi elemen x jika ditemukan.}

dictionary
    current: address
    position: integer

algorithm
    current <- L.first
    position <- 1

    // melakukan perulangan selama current tidak bernilai nil dan
    // info dari current tidak sama dengan x
    while ... do
        // seiring elemen current bergerak, position bertambah
        ...
        // lakukan perpindahan current
        ...
    endwhile
```

```

// cek jika current tidak bernilai nil
if ... then
    // tampilkan alamat dari current dan nilai dari posisi
    output(...)
else
    // tampilkan pesan "elemen tersebut tidak ada dalam list"
    output(...)
endif
endprocedure

```

Catatan: Gunakan pendekatan linier search untuk mencari elemen.

3. Mengurutkan List menggunakan Selection Sort

Buatlah prosedur **selectionSort** yang mengizinkan pengguna untuk mengurutkan elemen-elemen dalam list dari nilai terkecil ke terbesar.

```

procedure selectionSort(in/out L : list)
{I.S. Terdefinisi List L yang tidak kosong.
F.S. elemen pada list urut membesar berdasarkan infonya.}

dictionary
    p, min, temp: address
    x: infotype

algorithms
    p <- L.first

    // melakukan perulangan selama p tidak bernilai nil
    while ... do
        min <- p
        temp <- p

        // melakukan perulangan selama temp tidak bernilai nil
        while ... do
            // cek jika info dari temp lebih kecil dari min
            if ... do
                // ubah nilai min menjadi temp
                ...
            endif
            // lakukan perpindahan temp
            ...
        endwhile

        // tukar nilai info dari elemen p dan min, dengan bantuan x
        ...

        // lakukan perpindahan p
        ...
    endwhile
endprocedure

```

4. Menambahkan Elemen secara Terurut

Buatlah prosedur `insertSorted` menambahkan elemen baru ke dalam list yang sudah terurut.

```
procedure insertSorted(in/out L : list, in p : address)
{I.S. List tidak kosong.
 F.S. Menambahkan elemen p secara terurut.}

dictionary
    q, prev: address
    found: boolean

algorithms
    q <- L.first
    found <- false

    // melakukan perulangan selama q tidak bernilai nil dan found
    // masih false
    while ... do
        // cek jika info dari q lebih kecil dari p
        if ... then
            // mengisi prev dengan elemen q, dan lakukan perpindahan q
            ...
        else
            // mengisi found dengan nilai 'true'
            ...
        endif
    endwhile

    // melakukan pengecekan apakah prev bernilai nil
    if ... then
        // tambahkan p sebagai elemen pertama pada list
        ...
    else
        // tambahkan p pada posisi diantara prev dan q
        ...
    endif
endprocedure
```