

# Soal Assessment CLO 2 [Stack dan Queue]

## Struktur Data

### A. STACK (60 Poin)

Proses UNDO dan REDO adalah dua proses yang sering kita lakukan sebagai pengguna aktif *word processor*. Implementasikan kedua proses tersebut dalam representasi stack.

ADT dari stack UNDO dan REDO adalah sebagai berikut: (5 Poin)

#### Kamus

constant nMax: integer = 10

constant nil: integer = 0

type infotype: string

type stack: <tabS: array [1..nMax] of infotype,  
                  top: integer>

undoS, redoS: stack

#### 1. Cek apakah stack kosong (5 Poin)

function isEmpty(S: stack) → boolean

{ I.S S adalah stack kosong atau isi

F.S fungsi mengembalikan true jika S kosong dan false jika tidak  
kosong }

#### 2. Cek apakah stack penuh (5 Poin)

function isFull(S: stack) → boolean

{ I.S S adalah stack kosong atau isi

F.S fungsi mengembalikan true jika S penuh dan false jika tidak  
penuh }

#### 3. Buat stack kosong (5 Poin)

procedure createS(output S: stack)

{ I.S -

F.S S adalah stack kosong }

#### 4. Tambah sebuah elemen baru (5 Poin)

procedure push(input/output S: stack, input X: infotype)

{ I.S S adalah stack kosong atau isi. X adalah elemen baru yang  
akan ditambahkan ke S

F.S X adalah elemen teratas dari S }

#### 5. Hapus sebuah elemen (5 Poin)

procedure pop(input/output S: queue, output X: infotype)

{ I.S S adalah stack tidak kosong

F.S Elemen teratas terhapus; infonya dipegang oleh X; S mungkin  
kosong }

## Soal Assessment CLO 2 [Stack dan Queue]

### Struktur Data

#### 6. Buat prosedur *undo* (10 Poin)

procedure undo(input/output undoS: stack, output redoS: stack)  
{ I.S. undoS adalah stack berisi tumpukan aksi. redoS adalah stack  
berisi tumpukan aksi yang di-undo.  
F.S Elemen teratas undoS terhapus. Elemen yang terhapus menjadi  
elemen teratas redoS }

#### 7. Buat prosedur *redo* (10 Poin)

procedure redo(input/output redoS: stack, output undoS: stack)  
{ I.S. redoS adalah stack berisi tumpukan aksi yang di-undo. undoS  
adalah stack berisi tumpukan aksi  
F.S Elemen teratas redoS terhapus. Elemen yang terhapus menjadi  
elemen teratas undoS }

#### 8. Buat program main untuk mengimplementasikan prosedur/fungsi yang sudah dibuat (10 Poin)

### B. QUEUE (40 Poin)

Antrian mata kuliah yang akan dicoba untuk perkuliahan onsite akan disimpan dalam sebuah struktur data queue.

#### Dictionary

type adr: pointer to elmQ  
type elmQ: <kodeMK: string,  
namaDosen:string,  
next: adr>  
type queue: <head, tail: adr>

5 Poin

procedure add (input/output Q: queue, input P: address)  
{I.S. Q mungkin kosong. P adalah elemen yang akan ditambahkan, next(P)= nil  
F.S. P menjadi elemen terakhir dari Q}

10 Poin

procedure showDaftarMK(input Q: queue)  
{I.S. Q berisi data mata kuliah, Q mungkin kosong  
F.S. daftar mata kuliah tampil di layar }

10 Poin

procedure createElm(input kodeMK, namaDosen: string, output P: adr)  
{I.S. Q kosong  
F.S. Q berisi daftar matakuliah yang tersedia }

5 Poin

Pada main program, buatlah queue kosong kemudian isi dengan N mata kuliah dari user. Kemudian tampilkan mata kuliah tersebut ke layar. (10 Poin)