



**FAKULTAS TEKNIK**  
**UNIVERSITAS NEGERI YOGYAKARTA**  
**JOB SHEET ALGORITMA PEMROGRAMAN**

Semester : 2	Stack dan Queue	200 menit	
No. : LST/PTI/PTI 212/13	Revisi : 00	Tgl. : 28-4-2008	Hal. 1 dari 5 hal.

1. Kompetensi  
Mengenal dan memahami struktur data Stack (tumpukan) dan Queue (antrian).
2. Sub Kompetensi  
Dapat menyelesaikan suatu masalah dengan membuat dan mengaplikasikan struktur data Stack dan Queue ke dalam program.
3. Dasar Teori
  - A. Stack  
Stack adalah struktur data dalam pemrograman yang mempunyai sifat LIFO (Last In First Out).  
Operasi-operasi dalam Stack adalah :
    1. PUSH : digunakan untuk menambah data pada Stack (pada tumpukan paling atas).
    2. POP : digunakan untuk mengambil data pada Stack (pada tumpukan paling atas).
    3. ISEMPY : fungsi yang digunakan untuk mengecek apakah Stack masih kosong.
    4. ISFULL : fungsi yang digunakan untuk mengecek apakah Stack sudah penuh.
    5. CLEAR : digunakan untuk mengosongkan Stack.
  - B. Queue  
Queue adalah struktur data dalam pemrograman yang mempunyai sifat FIFO (First In First Out).  
Operasi-operasi dalam Queue adalah :
    1. ENQUEUE : digunakan untuk menambah data pada Queue (pada urutan paling belakang (ekor)).
    2. DEQUEUE : digunakan untuk mengambil data pada Queue (pada urutan paling depan (kepala)).
    3. ISEMPY : fungsi yang digunakan untuk mengecek apakah Queue masih kosong.
    4. ISFULL : fungsi yang digunakan untuk mengecek apakah Queue sudah penuh.
    5. CLEAR : digunakan untuk mengosongkan Queue.
4. Alat / Instrument / Aparatus / Bahan
5. Keselamatan Kerja

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------



**FAKULTAS TEKNIK**  
**UNIVERSITAS NEGERI YOGYAKARTA**  
**JOB SHEET ALGORITMA PEMROGRAMAN**

Semester : 2	Stack dan Queue	200 menit
No. : LST/PTI/PTI 212/13	Revisi : 00	Tgl. : 28-4-2008
Hal. 2 dari 5 hal.		

6. Langkah Kerja

A. Stack

1. Tentukan banyak Stack (misal : **const max = 10**).
2. Deklarasikan Stack dalam bentuk Record dimana didalamnya mengandung data dalam bentuk array dan suatu variabel inisialisasi untuk memantau posisi terakhir dari data yang ada dalam Stack.

```
tumpuk = record  
isi : array[1..max] of string;  
top : byte;  
end;
```

3. Deklarasikan variabel untuk Stack (**data : tumpuk**).
4. Buat prosedur untuk meng-inisialisasi Stack (juga merupakan CLEAR).

**Cara 1 :**

```
procedure inisialisasi;  
begin  
  data.top := 0;  
end;
```

**Cara 2 :**

```
procedure inisialisasi(var stack : tumpuk);  
begin  
  stack.top:=0  
end;
```

5. Buat fungsi ISFULL.

**Cara 1 :**

```
function is_full : boolean;  
begin  
  if data.top = max then is_full := true  
  else is_full := false;  
end;
```

**Cara 2 :**

```
function is_full (var stack : tumpuk) : boolean;  
begin  
  is_full := (stack.top = max)  
end;
```

6. Buat fungsi ISEMPY.

**Cara 1 :**

```
function is_empty : boolean;  
begin  
  if data.top = 0 then is_empty := true  
  else is_empty := false;  
end;
```

**Cara 2 :**

```
function is_empty (var stack : tumpuk) :  
boolean;  
begin  
  is_empty := (stack.top = 0)  
end;
```

7. Buat prosedur PUSH.

**Cara 1 :**

```
procedure push (dat : string);  
begin  
  if is_full = false then  
  begin  
    data.top := data.top + 1;  
    data.isi[data.top] := dat;  
  end;  
end;
```

**Cara 2 :**

```
procedure push (var stack : tumpuk;  
dat : string);  
begin  
  if not(is_full(stack)) then  
  begin  
    stack.top := stack.top + 1;  
    stack.isi[stack.top] := dat;  
  end;  
end;
```

Dibuat oleh :  
ADI

Dilarang memperbanyak sebagian atau seluruh isi dokumen  
tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh :



**FAKULTAS TEKNIK**  
**UNIVERSITAS NEGERI YOGYAKARTA**  
**JOB SHEET ALGORITMA PEMROGRAMAN**

Semester : 2	Stack dan Queue	200 menit
No. : LST/PTI/PTI 212/13	Revisi : 00	Tgl. : 28-4-2008
Hal. 3 dari 5 hal.		

8. Buat prosedur POP.

Misal :

<b>Cara 1 :</b> <pre>procedure pop (dat : string); begin   if is_empty = false then   begin     write(data.isi[data.top], ' diambil !');     data.top := data.top - 1;   end; end;</pre>	<b>Cara 2 :</b> <pre>procedure pop (var stack : tumpuk); begin   if not(is_empty(stack)) then   begin     writeln(stack.isi[stack.top], ' diambil !');     stack.top := stack.top - 1;   end; end;</pre>
---	---

9. Buat prosedur untuk menampilkan isi Stack.

Misal :

<b>Cara 1 :</b> <pre>procedure tampil; begin   for i := data.top downto 1 do     writeln(data.isi[i]); end;</pre>	<b>Cara 2 :</b> <pre>procedure tampil(var stack : tumpuk); begin   for i := stack.top downto 1 do     writeln(stack.isi[i]); end;</pre>
--	--

B. Queue

1. Tentukan banyak Queue (misal : **const max = 10**).
2. Deklarasikan Queue dalam bentuk Record dimana didalamnya mengandung data dalam bentuk array dan dua buah variabel inisialisasi untuk memantau posisi pertama dan terakhir dari data yang ada dalam Queue.

<b>Queue :</b> <pre>antrian = record   isi   : array[1..max] of string;   ekor  : byte; end;</pre>	<b>Circular Queue :</b> <pre>antrian=record   isi       : array[1..5] of char;   kepala,ekor : byte; end;</pre>
---	--

3. Deklarasikan variabel untuk Queue (**data : antrian**).

4. Buat prosedur untuk meng-inisialisasi Queue (juga merupakan CLEAR).

<b>Queue :</b> <pre>procedure inisialisasi(var queue : antrian); begin   queue.ekor := 0; end;</pre>	<b>Circular Queue :</b> <pre>procedure inisialisasi(var queue : antrian); begin   queue.kepala := max;   queue.ekor := max; end;</pre>
---	---

Dibuat oleh :  
ADI

Dilarang memperbanyak sebagian atau seluruh isi dokumen  
tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta

Diperiksa oleh :



**FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
JOB SHEET ALGORITMA PEMROGRAMAN**

Semester : 2	Stack dan Queue	200 menit
No. : LST/PTI/PTI 212/13	Revisi : 00	Tgl. : 28-4-2008
Hal. 4 dari 5 hal.		

5. Buat fungsi ISFULL.

<b>Queue :</b> <pre>function is_full(var queue : antrian):boolean; begin   is_full := (queue.ekor = max) end;</pre>	<b>Circular Queue :</b> <pre>function is_full(var queue : antrian) : boolean; var next : byte; begin   if queue.ekor = max then next:=1   else next := queue.ekor + 1;   is_full := (next = queue.kepala); end;</pre>
--	--

6. Buat fungsi ISEMPY.

<b>Queue :</b> <pre>function is_empty(var queue : antrian)   : boolean;  begin   is_empty := (queue.ekor = 0) end;</pre>	<b>Circular Queue :</b> <pre>function is_empty(var queue : antrian) : boolean; begin   is_empty := (queue.kepala = queue.ekor); end;</pre>
---	---

7. Buat prosedur ENQUEUE.

<b>Queue :</b> <pre>procedure enqueue(var queue : antrian;   dat : string);  begin   if not(is_full(queue)) then   begin     queue.ekor := queue.ekor + 1;     queue.isi[queue.ekor] := dat;   end end;</pre>	<b>Circular Queue :</b> <pre>procedure enqueue(var queue : antrian;   data : char);  begin   if not(is_full(queue)) then   begin     if queue.ekor = max then queue.ekor:=1     else queue.ekor := queue.ekor + 1;     queue.isi[queue.ekor] := data;   end; end;</pre>
--	--

8. Buat prosedur DEQUEUE.

<b>Queue :</b> <pre>procedure dequeue(var queue : antrian); var i : byte; begin   if not(is_empty(queue)) then   begin     writeln('Data ',queue.isi[1],' diambil!!!');     for i:=1 to queue.ekor-1 do       queue.isi[i] := queue.isi[i+1];     queue.ekor := queue.ekor - 1;   end; end;</pre>	<b>Circular Queue :</b> <pre>procedure dequeue(var queue : antrian); begin   if not(is_empty(queue)) then   begin     if queue.kepala = max then       queue.kepala := 1     else queue.kepala := queue.kepala + 1;     writeln('Data ',queue.isi[queue.kepala],' diambil!!!');   end; end;</pre>
--	--

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	---	------------------



**FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
JOB SHEET ALGORITMA PEMROGRAMAN**

Semester : 2	Stack dan Queue	200 menit
No. : LST/PTI/PTI 212/13	Revisi : 00	Tgl. : 28-4-2008
Hal. 5 dari 5 hal.		

9. Buat prosedur untuk menampilkan isi Queue.

<p><b>Queue :</b></p> <pre>procedure tampil(var queue : antrian); var i : byte; begin   for i:=1 to queue.ekor do     writeln(queue.isi[i]); end;</pre>	<p><b>Circular Queue :</b></p> <pre>procedure tampil(var queue : antrian); var i : byte; begin   if queue.kepala &lt; queue.ekor then     begin       for i:=queue.kepala + 1 to queue.ekor do         writeln(queue.isi[i]);     end   else     begin       for i:=queue.kepala + 1 to max do         writeln(queue.isi[i]);       for i:=1 to queue.ekor do         writeln(queue.isi[i]);     end; end;</pre>
---	--

7. Bahan Diskusi

1. Buatlah program untuk memanggil prosedur dan fungsi untuk masing-masing Stack dan Queue diatas dengan ketentuan pemasukan dan pengambilan datanya bisa lebih dari satu!
2. Buatlah suatu aplikasi untuk antrian pasien dokter!

Ketentuan :

1. Aplikasi dapat menampilkan status dari pasien, yaitu apakah sedang diperiksa atau masih mengantri.
2. Antrian pasien dapat diubah-ubah urutannya, kecuali untuk pasien yang sedang diperiksa.
3. Gunakan pendeklarasian seperti kode dibawah ini :

```
info = record
  nama,status : string[10];
  alamat      : string[15];
end;
antrian = record
  isi          : array[1..max] of info;
  kepala,ekor : integer;
end;

var
  data : antrian;
  pasien : info;
  .....dst.....
```

8. Lampiran :

Dibuat oleh : ADI	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------	--	------------------