

# TUGAS PENDAHULUAN

## MODUL XII

### LINKED LIST II : QUEUE

Prepared by: Davyn Vallerian (13213074) – Golfin Ekatria (13213139)

25 Februari 2016

#### Pengenalan

#### Pengertian Linked List

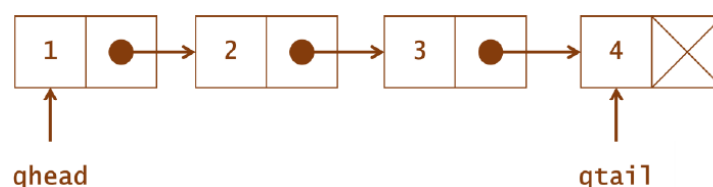


**Gambar 1** *Real-World Queue*

Pada Tugas Pendahuluan praktikum kali ini, anda tidak diminta untuk mengumpulkan program, melainkan mempersiapkan apa yang dibutuhkan dalam praktikum. **Anda tidak diperkenankan ikut praktikum apabila tidak mengerjakan tugas pendahuluan ini.** Asisten akan memeriksa pekerjaan Anda di awal praktikum.

Queue adalah salah satu struktur data bentukan yang memodelkan sebuah antrian. Ujung antrian (paling depan) disebut dengan *head of queue* sedangkan pangkal antrian (paling belakang) disebut *tail of queue*. Queue dimodelkan dengan linked list, namun dengan sedikit perbedaan pada pointernya.

Untuk lebih jelasnya, model queue dengan linked list dapat dilihat pada gambar berikut:

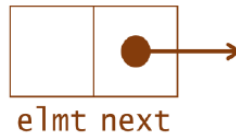


**Gambar 2** Ilustrasi Queue dalam Linked List

## Problem 1 : Implementasi Node dalam Data Structure

### *Definisi Masalah*

Sebagai persiapan untuk Tugas Praktikum, praktikan diminta untuk mendesain sebuah tipe bentukan yang merupakan **sebuah elemen dari Queue**, yang selanjutnya akan disebut sebagai *node*. Adapun sebuah *node* akan memiliki informasi berupa 1 buah data atau lebih dan sebuah *linker* menuju *node* yang lain. Perhatikan Ilustrasi berikut:



**Gambar 3** Ilustrasi Sebuah Node dalam Linked List

Untuk memudahkan praktikan, telah disediakan *hint code* sebagai berikut:

```
typedef struct node {
    /** Data Variables **/
    /* ... */
    /** Node Linker **/
    /* ... */
} node;
```

## Problem 2 : Implementasi Queue dalam Data Structure

### *Definisi Masalah*

Setelah kita mengimplementasikan tipe bentukan untuk satu element atau *node* dari queue, maka selanjutnya Anda diminta untuk membuat tipe bentukan untuk **queue** itu sendiri. Ilustrasi dapat dilihat pada gambar 2 di atas. Pointer yang menunjuk ke elemen paling depan dari queue diberi nama **qhead** sedangkan yang menunjuk elemen paling belakang diberi nama **qtail**.

Untuk memudahkan praktikan, telah disediakan *hint code* sebagai berikut:

```
typedef struct typeQueue {
    /** qHead Pointer **/
    /* ... */
    /** qTail Pointer **/
    /* ... */
} typeQueue;
```

## Problem 3 : Implementasi Prosedur pushTail pada Queue

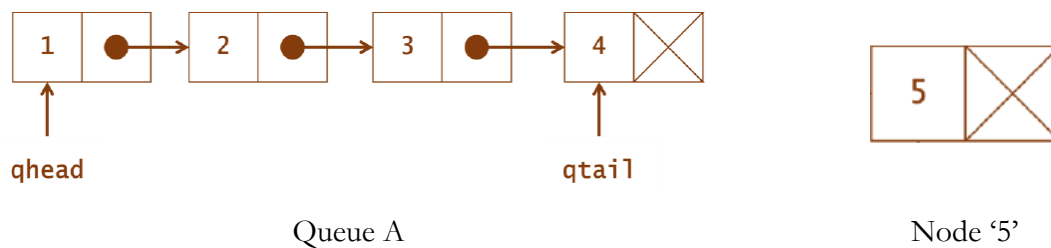
### *Definisi Masalah*

```
void pushTail (int Data, typeQueue *queue);
```

Dengan parameter di atas, buatlah prosedur **pushTail**. Prosedur ini akan menambahkan satu buah *node* di belakang *node Tail* dari sebuah Queue yang dituju, menjadikan *node* yang akan diinput ini sebuah *Tail* yang baru bagi Queue tersebut.

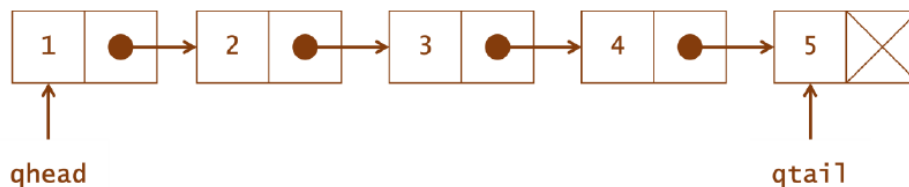
*Agar dapat lebih mudah dimengerti, diberikan ilustrasi sebagai berikut:*

Akan dicoba untuk menaruh *Node '5'* pada Queue A.



**Gambar 4** Queue dan Node sebelum `pushTail` dijalankan.

Dimana ketika `pushTail` dijalankan:



**Gambar 5** Queue dan Node setelah `pushTail` dijalankan.

Langkah-langkah penting yang harus dilakukan dalam prosedur `pushTail`:

1. *Link node Tail* dari Queue pada *node* yang di-`pushTail`-kan.
2. Pindahkan *pointer qTail* pada *node* yang baru dimasukkan.

#### Problem 4 : Implementasi Fungsi `popHead` pada Queue

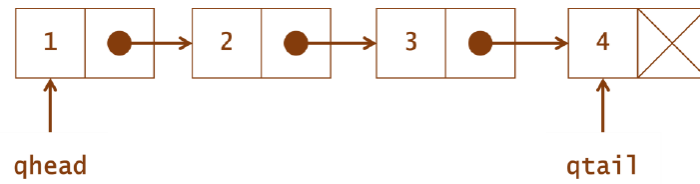
##### *Definisi Masalah*

```
int popHead (typeQueue *queue);
```

Dengan parameter di atas, buatlah fungsi `popHead`. Fungsi ini akan mengambil data yang tersimpan pada bagian *head* dari suatu Queue dan memberikan *return value* sesuai dengan data tersebut. Fungsi kemudian akan melepaskan bagian *head* dari Queue, menjadikan *node* yang *ter-linked* dengan *head* yang akan dilepaskan menjadi *head* yang baru.

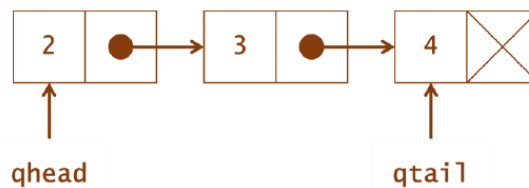
Agar dapat lebih mudah dimengerti, diberikan ilustrasi sebagai berikut:

Akan dicoba untuk menjalankan fungsi popHead Queue A.



**Gambar 6** Queue sebelum popHead dijalankan.

Dimana ketika popHead dijalankan:



**Gambar 7** Queue setelah popHead dijalankan.

Dan pada *return*, fungsi akan mengembalikan nilai **1**.

Langkah-langkah penting yang harus dilakukan dalam prosedur popHead:

1. Ambil data yang tersimpan pada *node head Queue*.
2. Pindahkan *pointer qHead* pada *node* yang berada setelah *Head*.

### Petunjuk Penyerahan Tugas Pendahuluan Modul XII

Buatlah *base code* seperti berikut:

```
/****** Header *****/

/****** Type Defining *****/
typedef struct node {
    /** Data Variables **/
    /* ... */
    /** Node Linker **/
    /* ... */
} node;

typedef struct typeQueue {
    /** qHead Pointer **/
    /* ... */
    /** qTail Pointer **/
    /* ... */
} typeQueue;

/****** Function & Procedure Declaration *****/
void pushTail (int Data, typeQueue *queue);
int popHead (typeQueue *queue);
```

```

/***** Main Program *****/
int main () {
    /*** Variables ***/
    /* ... */
    /*** Algorithm ***/
    /* ... */
}

/***** Function & Procedure Algorithm *****/
void pushTail (int Data, typeQueue *queue) {
    /* ... */
}

int popHead (typeQueue *queue) {
    /* ... */
}

```

Tunjukkan kepada asisten *base program* ini sebagai syarat mengikuti praktikum.

**+ End +**

