

TUGAS PENDAHULUAN

MODUL XI

Queue

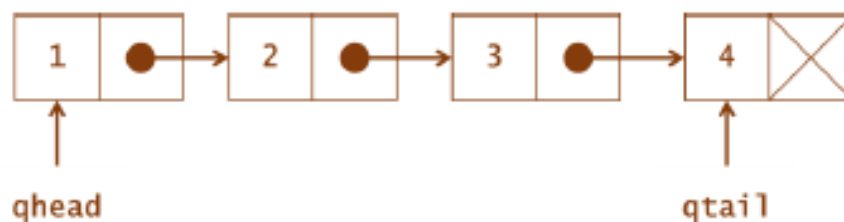
Pada modul ini, *library* yang digunakan **hanya** `stdio.h`. Apabila membutuhkan fungsi matematika pada `library math.h`, maka buatlah fungsi tersebut menjadi **fungsi/prosedur baru**.

Pengenalan

Pengertian Queue

Pada praktikum sebelumnya sudah dikenalkan tentang struktur data, dan linked list. Queue, pada aplikasinya menggunakan struktur data dan linked list.

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 pointer-nya. Untuk lebih jelasnya, model queue dengan linked list dapat dilihat pada gambar berikut:



Gambar 1: Implementasi queue dengan linked list

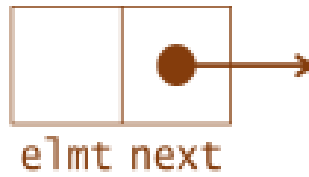
Pada queue, fungsi untuk menambahkan elemen ke dalam antrian disebut **tail()**, sedangkan untuk mengambil elemen paling depan disebut **head()**.

Pada tugas pendahuluan ini, Anda diminta untuk mengimplementasikan tipe data bentukan untuk **queue** dan juga implementasi fungsi **head()** dan **tail()**. Semuanya akan digunakan di dalam praktikum.

Problem 1 : Implementasi Struktur Data Element Linked List

Definisi Masalah

Untuk mempermudah pekerjaan kita ke depan, kita harus membuat tipe bentukan untuk **satuelement dari queue**. Pada praktikum kali ini, satu node terdiri dari dua buah data dan satu penunjuk. Perhatikan ilustrasi berikut



Gambar 2: Contoh satu element (*node*) queue

Tipe bentukan untuk satu element queue ini, kita sebut sebagai **node**. Data di dalamnya diberi nama **elmt** dan pointer ke node selanjutnya diberi nama **next**. Implementasikan tipe bentukan untuk *node*.

Berikut adalah kode bantu:

```
typedef struct node
{
    /** Isi apa yang kurang
    Pada struktur ini
    ...
    ... **/
}node;
```

Problem 2 : Implementasi Struktur Data Queue

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**.

Diberikan bantuan kode sebagai berikut:

```
typedef struct queue_type
{
    /** Isi apa yang kurang
    Pada struktur ini
    ...
    ... **/
}queue_type;
```

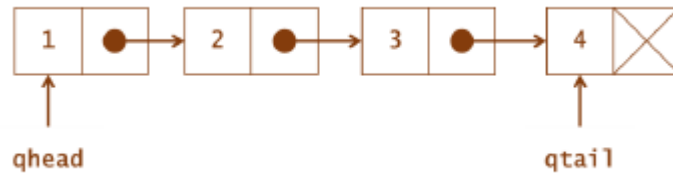
Problem 3 : Implementasi Fungsi Tail

Definisi Masalah

Bentuk prototype dari fungsi `push()` umumnya adalah sebagai berikut ini:

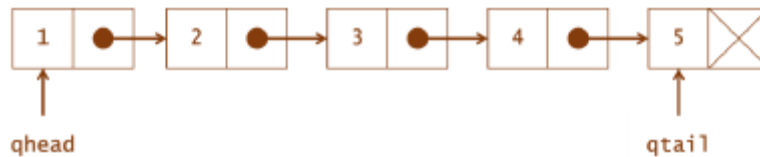
```
Void tail(int elmt , queue_type *queue);
```

Fungsi `tail` akan menambahkan element ke akhir queue. Misalkan sebelum fungsi dipanggil, kita memiliki queue sebagai berikut:



Gambar 3: Sebelum taildijalankan

Kemudian apabila kita memanggil fungsi tail dengan cara **tail(5, &queue)**, maka queue yang kita punya akan menjadi:



Gambar 5: Setelah taildijalankan

Implementasikanlah fungsi tail() ini, hingga bekerja dengan baik.

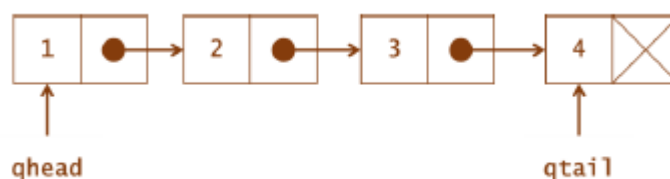
Problem 4 : Implementasi Fungsi Head

Definisi Masalah

Kode prototype dari fungsi pop adalah

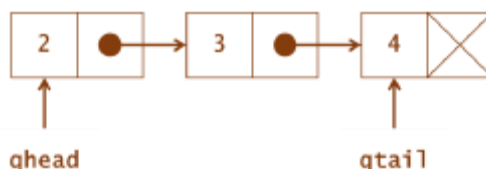
```
Void head( queuetype *queue);
```

Fungsi head akan mengambil elemen pertama dari queue dan menjadikan nilainya sebagai return fungsi. Misalkan sebelum fungsi dipanggil, kita memiliki list sebagai berikut:



Gambar 4: Sebelum head dijalankan

Kemudian apabila kita memanggil fungsi head dengan cara **head(&queue)**, maka queue yang kita punya akan menjadi:



Gambar 5: Setelah head dijalankan

Selain mengubah queue, fungsi ini juga mengembalikan (*return*) nilai 1 pada contoh di atas. Implementasikanlah fungsi head() ini, hingga bekerja dengan baik.

Implementasikanlah fungsi push() ini, hingga bekerja dengan baik.

Dari keseluruhan tugas diatas, buatlah sebuah program queue sederhana, dan tampilkan isi queue.

DILARANG menggunakan struktur data **array**

Deliverable

Buatlah keempat tugas diatas dalam satu program, disimpan dalam bentuk filetpe .c. simpan file dengan nama TP_Problem1234.c. Jangan lupa memberikan identitas (header file) di awal file ini. Pastikan program dapat dikompilasi dan dijalankan dengan benar.

File yang akan dikirim akan memiliki susunan kurang lebih seperti di bawah ini:

```
/* ... library, definition, etc */

typedef struct node
{
    /** isi apa yang kurang
    Pada struktur ini
    ... **/
} node;

typedef struct queue_type
{
    /** isi apa yang kurang
    Pada struktur ini
    ... **/
} queue_type;

/** prototype fungsi **/
void tail(double elmt, queue_type *queue);
void head(queue_type *queue);

int main(void)
{
    /** fungsi utama **/
}

/** implementasi fungsi **/
void tail(double elmt, queue_type *queue)
{
    /** implementasi fungsi tail **/
}

void head(queue_type *queue)
{
    /** implementasi fungsi head **/
}

void head(queue_type *queue)
{
    /** implementasi fungsi untuk menampilkan isi queue **/
}
```

Petunjuk Penyerahan Tugas Pendahuluan Modul XI

Simpan file (TP_Problem1234.c) dalam satu folder. Gunakan program WinRAR untuk mengkompresi menjadi arsip .Zip. Penamaan file Zip menggunakan format yang telah diberikan. File .Zip ini yang akan di-submit melalui Google Classroom Praktikum PMC 2019. Hanya file kode saja yang dimasukkan ke dalam arsip .Zip. File executable tidak perlu dimasukkan..

Selesai