

## MODUL 8: Antrian Berprioritas

---

### 1.1. Deskripsi Singkat

Queue atau antrian adalah suatu struktur data berbentuk list atau daftar yang proses pengelolaan datanya menggunakan konsep FIFO(First In First Out) atau LILO(Last In Last OUT). Antrian berprioritas adalah himpunan elemen, yang setiap elemennya telah diberikan sebuah prioritas. Elemen pada antrian berprioritas harus menyimpan nilai prioritas pada setiap elemen. Untuk memasukan elemennya, tidak harus melalui ujung rear, tetapi disisipkan dan diurutkan berdasarkan prioritas elemen.

### 1.2. Tujuan Praktikum

- 1) Mahasiswa mampu mengimplementasikan berbagai persoalan antrian prioritas dengan menggunakan bahasa C
- 2) Mahasiswa dapat melakukan operasi penyesipan (Enqueue) maupun penghapusan (Dequeue) item dari antrian prioritas dengan menggunakan bahasa C

### 1.3. Material Praktikum

Kegiatan pada modul ini memerlukan material berupa software editor dan compiler (atau IDE) untuk bahasa pemrograman C.

### 1.4. Kegiatan Praktikum

Pada praktikum kali ini, kita akan membuat sebuah program antrian berprioritas yang akan memilih 5 orang perwakilan mahasiswa yg akan mengikuti lomba pemrograman. Kelima orang perwakilan ini dipilih pertama berdasarkan nilai UTS Alpro, kemudian berdasarkan nilai UTS Kalkulus dan terakhir berdasarkan urutan pendaftaran.

Dalam proses pembuatannya, ada 4 tahap, yaitu:

1. Persiapan,
2. Membuat fungsi dequeue,
3. Membuat fungsi enqueue,
4. Finalisasi.

#### A. Persiapan

Dalam tahapan ini kita harus mempersiapkan struktur data yang dibutuhkan serta fungsi lain yang akan digunakan dalam proses berikutnya.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Node
typedef struct node {
    char nama[20];
    int alpro;
    int kalkulus;
    struct node* next;
} mhs;

int count=0;

// Function to Create A New Node
mhs* newmhs(char a[], int alp, int kal)
{
    mhs* temp = (mhs*)malloc(sizeof(mhs));
    strcpy(temp->nama, a);
    temp->alpro = alp;
    temp->kalkulus = kal;
    temp->next = NULL;

    return temp;
}

```

sampai dengan potongan program di atas, kita sudah menyiapkan struktur data serta beberapa fungsi yang nanti akan digunakan dalam proses selanjutnya.

ketik potongan program tersebut pada IDE Anda, lalu simpan dengan nama **praktikum8.c**.

## B. Pembuatan Fungsi dequeue

Untuk membuat fungsi dequeue, ada dua kondisi yang harus diperhatikan. Kondisi pertama yaitu saat antrian masih kosong dan selain itu. Jika masih kosong maka tidak bisa menghapus, jika tidak baru menghapus antrian dari depan.

```
// menghapus pendaftar
void dequeue(mhs** head)
{
    if ((*head) != NULL) {
        mhs* temp = *head;
        (*head) = (*head)->next;
        free(temp);
    }
}
```

### C. Pembuatan Fungsi enqueue

Fungsi enqueue adalah fungsi utama dari program ini. Ada beberapa kondisi yang perlu diperhatikan berkaitan dengan penempatan antrian baru.

```
void enqueue(mhs** head, char n[], int alp, int kal)
{
    mhs* temp = newmhs(n, alp, kal);

    if ((*head) == NULL) {
        (*head) = temp;
    }

    else if (alp > (*head)->alpro) {
        temp->next = *head;
        (*head) = temp;
    }

    else{
        if (kal > (*head)->kalkulus) {
            temp->next = *head;
            (*head) = temp;
        }

        else{
            mhs*start=(*head);
```

```

        while (start->next != NULL && start->next-
>kalkulus > kal) {
            start = start->next;
        }

        temp->next = start->next;
        start->next = temp;
    }
}

count++;
}

```

#### D. Finalisasi

Langkah terakhir adalah mengatur fungsi-fungsi tersebut sehingga antrian bisa menyimpan hasil yang sesuai.

```

void display(mhs* head)
{
    if(head == NULL)
    {
        printf("Belum ada yang daftar\n");
    }
    else
    {
        printf("Nama:%s Alpro: %d Kalkulus: %d\n", head-
>nama, head->alpro, head->kalkulus);
        display(head->next);
    }
}

int main()
{
    mhs* wakil=NULL;

    enqueue(&wakil, "Eko", 50,20);
}

```

```
enqueue(&wakil, "Budi", 50,20);
enqueue(&wakil, "bambang", 60,20);
enqueue(&wakil, "Eka", 60,20);
enqueue(&wakil, "wawo", 60,20);
enqueue(&wakil, "Ame", 60,30);

display(wakil);

return 0;
}
```

### 1.5. Penugasan

1. Modifikasi program agar secara otomatis antrian hanya maksimal berisi 5 pendaftar terbaik sesuai kriteria yang ditentukan.
2. Modifikasi program tersebut agar menampilkan mulai dari urutan terbawah.
3. Buat program versi array.