

MODUL 7: Antrian

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 LIFO (Last In Last Out), sehingga proses menambah data dan penghapusannya berbeda dengan tumpukan, menggunakan dua pintu. Menambahkan data ke rear atau tail dan menghapus data mulai dari head atau front. Antrian dapat diimplementasikan dengan array atau linked list.

1.2. Tujuan Praktikum

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

A. Contoh Kasus Penggunaan Queue

Pada praktikum kali ini, kita akan membuat sebuah program menggunakan queue untuk generate 1 sampai N bilangan biner.

Contoh:

N = 10

Angka Binary: 1 10 11 100 101 110 111 1000 1001 1010 (lihat pola perubahan angka-nya)

Dalam proses pembuatannya, kita harus mempersiapkan struktur data yang dibutuhkan serta fungsi lain yang akan digunakan dalam proses berikutnya, seperti:

- fungsi generate bilangan biner,
- fungsi enqueue,
- dan fungsi dequeue.

1. Pertama-tama, ketik potongan program di bawah ini pada IDE Anda, lalu simpan dengan nama **praktikum7a.c**.

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

#define MAX 50
#define MAX_CHARS 16
char QUEUE[MAX_CHARS][MAX];
int rear, front;
```

Sampai dengan potongan program di atas, kita sudah menyiapkan library, struktur data, dan variabel-variabel yang akan digunakan dalam proses pembuatan antrian selanjutnya. Kita juga sudah mendeklarasikan antrian berupa array bertipe **char**.

2. Berikutnya, tambahkan fungsi generate bilangan biner pada **praktikum7a.c**, seperti yang tertera di bawah ini.

```
void generateBinaryNumbers(int N)
{
    char bin[] = "1";
    enqueue(bin);

    for(int i=0; i < N; i++)
    {
        char temp[MAX_CHARS] = "";
        char temp2[MAX_CHARS] = "";

        strcpy(temp, QUEUE[front]);
        strcpy(temp2, QUEUE[front]);

        printf("%s ", temp);
        dequeue();

        strcat(temp, "0");
        enqueue(temp);

        strcat(temp2, "1");
        enqueue(temp2);
    }
}
```

Pahami algoritme pada fungsi `generateBinaryNumbers` di atas. Jelaskan cara kerja fungsi tersebut!

3. Sebagai fungsi utama dari program ini, yang berguna untuk menyimpan rangkaian bilangan biner yang digenerate, kita buat sebuah fungsi enqueue. Perhatikan bahwa terdapat beberapa kondisi yang perlu divalidasi sebelum penempatan antrian baru.

```
void enqueue(char item[])
{
    if(rear == MAX - 1){
        printf("Antrian penuh \n");
        return;
    }

    if(front == -1)
        front = 0;

    rear++;
    //karena antrian bertipe char, maka enqueue tidak bisa
    //dilakukan dengan cara QUEUE[rear]=item
    strcpy(QUEUE[rear], item);
}
```

4. Selanjutnya, untuk membuat fungsi dequeue, ada kondisi yang juga harus diperhatikan, yaitu antrian masih kosong atau sudah ada isi. Jika masih kosong, maka dequeue tidak bisa dilakukan. Jika antrian sudah ada isinya, maka dequeue/menghapus antrian bisa dilakukan dari depan.

```
void dequeue()
{
    if(rear == -1){
        printf("Antrian kosong \n");
        return;
    }

    if(front == rear)
        front = rear = -1;
    else{
        for(int i = 0; i < rear; i++) {
            strcpy(QUEUE[i], QUEUE[i + 1]);
        }
    }
}
```

```

        rear--;
        front = 0;
    }
}

```

5. Langkah terakhir, aturlah fungsi-fungsi di atas pada fungsi main, sehingga user bisa menginputkan jumlah bilangan biner yang ingin digenerate dan antrian bisa menyimpan hasil yang sesuai.

Pada fungsi main, Inisialisasi nilai rear dan front dengan nilai -1.

B. Antrian Berprioritas

kita akan membuat sebuah program antrian berprioritas untuk 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. Kita harus mempersiapkan struktur data yang dibutuhkan serta fungsi-fungsi lain yang akan digunakan dalam proses berikutnya.

1. Untuk menyelesaikan permasalahan di atas, kita siapkan struktur data berupa antrian dalam bentuk linked list. Ketik potongan program di bawah ini, lalu simpan dengan nama **praktikum7b.c**.

```

#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;
}

```

2. Berikutnya, kita lengkapi dengan fungsi dequeue dan enqueue berikut.

```

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

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++;
}

```

Pahami cara kerja fungsi-fungsi di atas, kemudian jelaskan.

3. 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;
}

```

Pahami dan jelaskan cara kerja fungsi main tersebut

1.5. Penugasan

1. Menggunakan **stack**, buat sebuah program untuk menentukan apakah input string yang berisi pasangan tanda kurung sudah benar/seimbang.

Contoh:

Input : [{()}]	Output : TRUE / "Benar" / 1
Input : [()]	Output : FALSE / "Salah" / 0
Input : [()]{ }{ [()] () }	Output : TRUE / "Benar" / 1

Simpan program dengan nama **praktikum7c_kelas_nim.c**.

2. Menggunakan **stack**, buatlah program untuk membalik karakter-karakter dalam suatu kata. (karakter depan menjadi belakang dan karakter belakang menjadi karakter depan, dan seterusnya) dengan menggunakan Stack. Sebagai contoh: "RUSAK" menjadi "KASUR".

Simpan program dengan nama **praktikum7d_kelas_nim.c**.

3. Buatlah sebuah program antrian dalam bentuk linked list, untuk antrian pasien di sebuah klinik. Menggunakan program tersebut, pasien dapat:
 - a. Mengambil nomor antrian berobat
 - b. Melihat antrian yang tersisa
 - c. Melihat perkiraan waktu tunggu pasien pada saat mengantri untuk berobat (Perkiraan waktu pemeriksaan per pasien 15 menit)
 - d. Keluar dari program

Simpan program dengan nama **praktikum7e_kelas_nim.c**.

Gabungkan ketiga file penugasan di atas menjadi sebuah file zip/rar. Format penamaan file zip/rara: **praktikum7_kelas_nim.zip**. Unggah file tersebut di Google Classroom sesuai dengan batas waktu yang telah ditetapkan.