

Nama: Muhammad Muhyi Akbar

Nim; 1203230090

IF 02

Source code

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define MAX 10

typedef struct
{
    int data[MAX];
    int front;
    int rear;
} Queue;

void init(Queue *queue)
{
    queue->front = queue->rear = -1;
}

bool isEmpty(Queue *queue)
{
    return queue->front == -1 && queue->rear == -1;
}

bool isFull(Queue *queue)
{
    return queue->rear == MAX - 1;
}

char *checkPatientSeverity(int patient)
{
    if (patient >= 1 && patient <= 5)
    {
        return "ditangani dokter umum";
    }
    else
    {
        return "ditangani dokter spesialis";
    }
    return "unknown";
}
```

```

void enqueue(Queue *queue, int patient)
{
    if (!isFull(queue)) // Jika queue tidak penuh
    {
        printf("%s\n", checkPatientSeverity(patient)); // Print hasil dari
        checkPatientSeverity
        if (isEmpty(queue)) // Jika queue kosong
        {
            queue->front++; // front di increment
            (ditambah 1) karena queue kosong (front = -1, rear = -1)
            queue->data[++queue->rear] = patient; // rear di increment
            (ditambah 1) dan data di rear diisi dengan patient
            return;
        }
        queue->data[++queue->rear] = patient; // rear di increment (ditambah
        1) dan data di rear diisi dengan patient
        return;
    }
    printf("Queue is full\n");
}

int main()
{
    Queue queue;
    int patient[10];
    init(&queue);

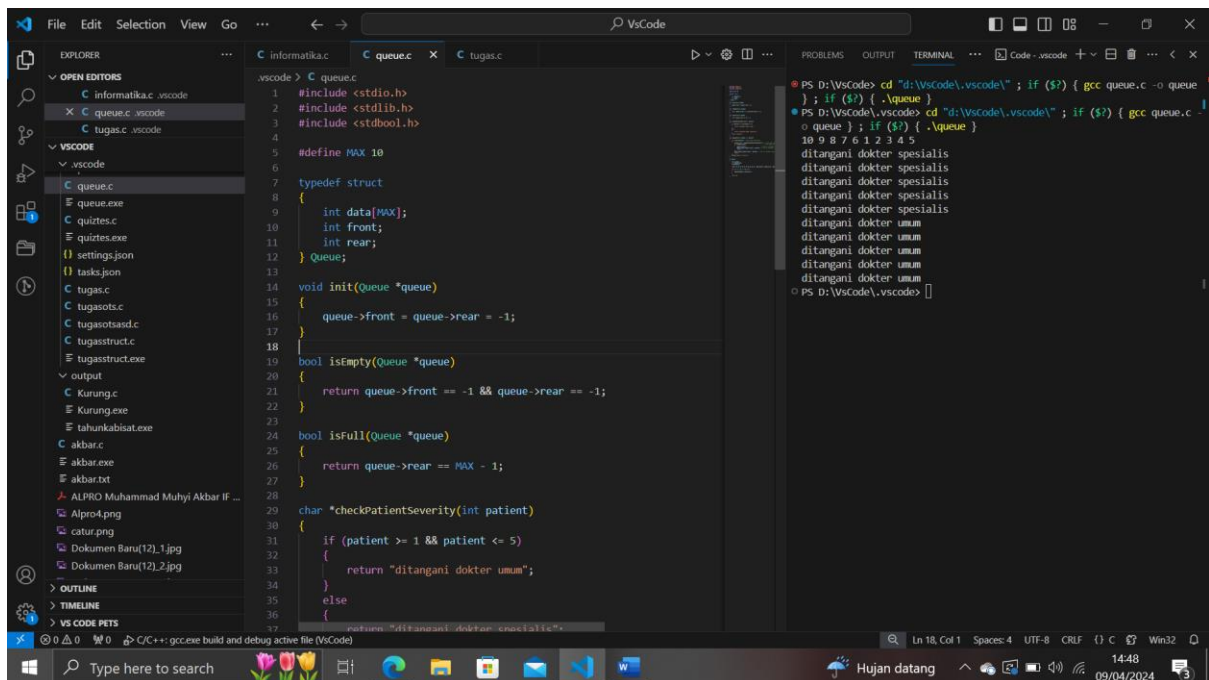
    scanf("%d %d %d %d %d %d %d %d %d %d", &patient[0], &patient[1],
    &patient[2], &patient[3], &patient[4], &patient[5], &patient[6], &patient[7],
    &patient[8], &patient[9]);

    for (int i = 0; i < 10; i++)
    {
        enqueue(&queue, patient[i]);
    }

    return 0;
}

```

Output



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4
5 #define MAX 10
6
7 typedef struct
8 {
9     int data[MAX];
10    int front;
11    int rear;
12 } Queue;
13
14 void init(Queue *queue)
15 {
16     queue->front = queue->rear = -1;
17 }
18
19 bool isEmpty(Queue *queue)
20 {
21     return queue->front == -1 && queue->rear == -1;
22 }
23
24 bool isFull(Queue *queue)
25 {
26     return queue->rear == MAX - 1;
27 }
28
29 char *checkPatientSeverity(int patient)
30 {
31     if (patient >= 1 && patient <= 5)
32     {
33         return "ditangani dokter umum";
34     }
35     else
36     {
37         return "ditangani dokter spesialis";
38     }
39 }
```

```
PS D:\VsCode> cd "d:\VsCode\vscode\" ; if ($?) { gcc queue.c -o queue } ; if ($?) { .\queue }
PS D:\VsCode\vscode> cd "d:\VsCode\vscode\" ; if ($?) { gcc queue.c } ; if ($?) { .\queue }
10 9 8 7 6 1 2 3 4 5
ditangani dokter spesialis
ditangani dokter spesialis
ditangani dokter spesialis
ditangani dokter spesialis
ditangani dokter umum
ditangani dokter umum
ditangani dokter umum
ditangani dokter umum
ditangani dokter umum
ditangani dokter umum
PS D:\VsCode\vscode>
```

Penjelasan

Program ini menggunakan struktur data antrian (queue) yang diimplementasikan dengan array.

Terdapat fungsi-fungsi dasar seperti init, isEmpty, isFull, dan enqueue untuk mengatur antrian.

Terdapat fungsi checkPatientSeverity yang menentukan apakah pasien akan ditangani oleh dokter umum atau spesialis berdasarkan tingkat keparahan penyakitnya.

Dalam main function, program meminta pengguna untuk memasukkan tingkat keparahan pasien.

Setelah itu, program memasukkan pasien ke dalam antrian dan menentukan dokter yang akan menanganinya berdasarkan tingkat keparahan penyakitnya.

Jika antrian penuh, program akan mencetak "Queue is full"