

```
#include <iostream>
#include <stdlib.h>

#define MAX 10

using namespace std;

void insert (int queue[], int *rear, int nilai);
void del (int queue[], int *front, int rear, int *nilai);

main()
{
    int queue[MAX];
    int front, rear;
    int n, nilai;

    front = rear = (-1);
    do
    {
        do
        {
            cout << "Masukkan Nilai Elemen : ";
            cin >> nilai;
            insert (queue, &rear, nilai);

            cout << endl;
            cout << "Tekan 1 untuk Melanjutkan : ";
            cin >> n;
        }while (n == 1);

        cout << endl;
        cout << "Tekan 1 untuk Menghapus Elemen : ";
        cin >> n;

        while (n == 1)
        {
            del (queue, &front, rear, &nilai);
            cout << "Nilai telah dihapus : " << nilai << endl;
            cout << endl;
            cout << "Tekan 1 untuk Menghapus Elemen : ";
            cin >> n;
        }

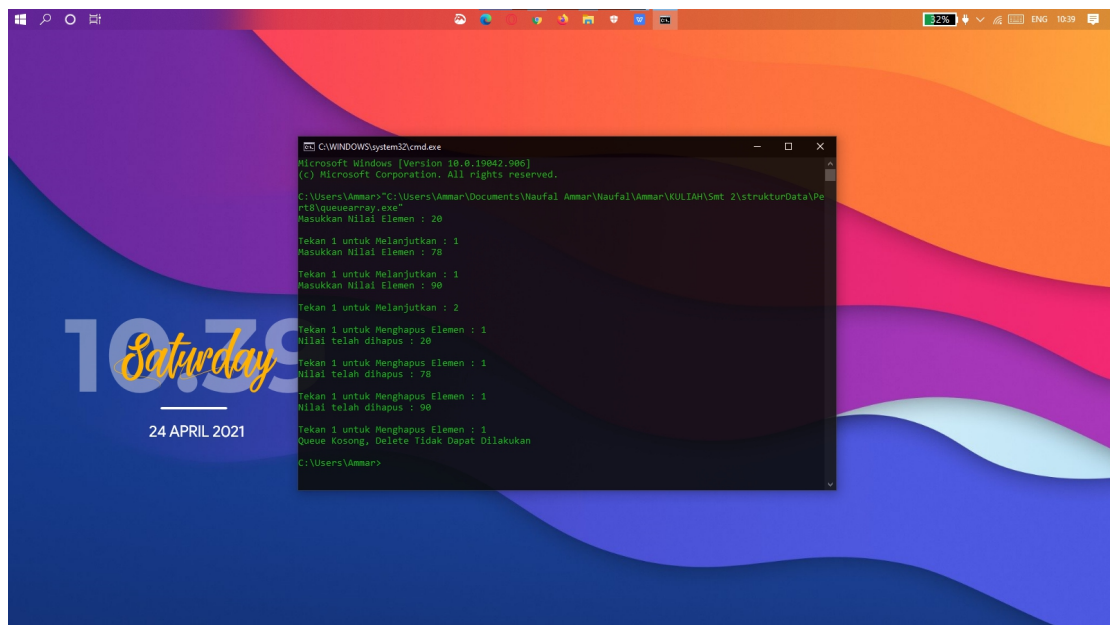
        cout << endl;
        cout << "Tekan 1 untuk Melanjutkan : ";
        cin >> n;
    } while (n == 1);
}

void insert (int queue[], int *rear, int nilai)
{
```

```
        if (*rear < MAX-1)
        {
            *rear = *rear + 1;
            queue[*rear] = nilai;
        }
        else
        {
            cout << "Queue Penuh, Insert Tidak Dapat Dilakukan" <<
endl;
            exit(0);
        }
    }

void del (int queue[], int *front, int rear, int *nilai)
{
    if (*front == rear)
    {
        cout << "Queue Kosong, Delete Tidak Dapat Dilakukan" <<
endl;
        exit(0);
    }

    *front = *front + 1;
    *nilai = queue[*front];
}
```



Naufal Ammar Hidayatulloh
2010631170104
2E Teknik Informatika

```
#include <iostream>
#include <stdlib.h>

#define Nil NULL

using namespace std;

struct node
{
    int data;
    struct node *link;
};

void insert(struct node **front, struct node **rear, int nilai)
{
    struct node *temp;
    temp = (struct node *)malloc(sizeof(struct node));

    if (temp == Nil)
    {
        cout << "Error, memori penuh" << endl;
        exit(0);
    }
    temp->data = nilai;
    temp->link = Nil;
    if (*rear == Nil)
    {
        *rear = temp;
        *front = *rear;
    }
    else
    {
        (*rear)->link = temp;
        *rear = temp;
    }
}

void del(struct node **front, struct node **rear, int *nilai)
{
    struct node *temp;

    if ((*front == *rear) && (*rear == Nil))
    {
        cout << "Queue Kosong, Delete Tidak Dapat Dilakukan" <<
endl;
        exit(0);
    }

    *nilai = (*front)->data;
    temp = *front;
    *front = (*front)->link;
```

```
        if (*rear == temp)
            *rear = (*rear)->link;
        free(temp);
    }

main()
{
    struct node *front = Nil, *rear = Nil;
    int n, nilai;

    do
    {
        do
        {
            cout << "Masukkan Nilai Elemen : ";
            cin >> nilai;
            cout << endl;
            insert(&front, &rear, nilai);
            cout << "Tekan 1 untuk Melanjutkan : ";
            cin >> n;
        } while (n == 1);

        cout << endl
             << "Tekan 1 untuk Menghapus Elemen : ";
        cin >> n;

        while (n == 1)
        {
            del(&front, &rear, &nilai);
            cout << endl
                 << "Nilai yang di Hapus : " << nilai << endl
                 << "Tekan 1 untuk Menghapus Elemen : ";
            cin >> n;
        }

        cout << endl
             << "Tekan 1 untuk Melanjutkan : ";
        cin >> n;
    } while (n == 1);
}
```

Naufal Ammar Hidayatulloh
2010631170104
2E Teknik Informatika

