

BAB III ANTRIAN DENGAN ARRAY

3.1 Pengertian Antrian

Antrian (Queue) dapat diartikan sebagai suatu kumpulan data yang seolah-olah terlihat seperti ada data yang diletakkan di sebelah data yang lain seperti pada gambar 01. Pada gambar, data masuk melalui lorong di sebelah kanan dan masuk dari terowongan sebelah kiri. Hal ini membuat antrian bersifat FIFO (First In First Out), beda dengan stack yang berciri LIFO.



Gambar 01

Antrian dapat dibuat baik dengan array maupun dengan struct. Pada pembuatan antrian dengan array, antrian yang disajikan bersifat statis. Ini disebabkan oleh jumlah maksimal array sudah ditentukan sejak deklarasi awal.

3.1.1 Algoritma PUSH

Salah satu algoritma untuk proses memasukkan data (PUSH) adalah sebagai berikut:

0. Masukkan inputan (x)
1. Jika variabel cek = max (Nilai maksimal array), kerjakan langkah 2. Jika tidak, kerjakan langkah 3.
2. Cetak "ANTRIAN PENUH" lalu selesai.
3. Selama cek kurang dari max, maka $c \leftarrow c + 1$ dan data $[c] \leftarrow x$.

3.1.2 Algoritma POP

Salah satu algoritma untuk proses mengeluarkan data (POP) adalah sebagai berikut:

0. Jika cek = 0, cetak "ANTRIAN KOSONG" kemudian selesai. Jika tidak, lakukan langkah 2.
1. mulai $x=0$, selama x kurang dari cek, lakukan langkah 2 dan 3.
2. data $[x] \leftarrow$ data $[x+1]$.
3. data $[cek-1] \leftarrow$ NULL.
4. cek \leftarrow cek - 1

Contoh program yang mengimplementasikan algoritma di atas :

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

void main()
{
    int cek=0, data[20], x, hapus;
    char pil;
    do {
        clrscr();
        printf("1. Tambah Antrian\n");
        printf("2. Hapus Antrian\n");
        printf("3. Lihat Antrian\n");
        printf("4. Keluar\n");
        printf("Silahkan masukkan pilihan anda... ");
        pil=getche();
```

```

if(pil!='1' && pil !='2' && pil !='3' && pil!='4' )
    printf("\n\nAnda salah mengetikkan inputan...\n");
else
{
    if(pil=='1')    //PUSH
    {
        if(cek==20)
            printf("\nAntrian Penuh\n\n");
        else
        {
            printf("\nMasukkan nilai--> ");scanf("%i",&x);
            data[cek]=x;
            cek++;
        }
    }
    else
    {
        if(pil=='2')    //POP
        {
            if(cek==0)
                printf("\nAntrian kosong\n\n");
            else
            {
                hapus=data[0];
                for(int v=0;v<cek;v++)
                    data[v]=data[v+1];
                data[cek-1]=NULL;
                cek--;
                printf("\nData dgn nilai=%i terhapus.",hapus);
            }
            getch();
        }
        else
        {
            if(pil=='3')    //CEK DATA
            {
                if(cek==0)
                    printf("\nAntrian Kosong.\n\n");

                else
                {
                    printf("\n");
                    for(int z=0;z<cek;z++)
                    {
                        printf(" | ");
                        printf("%i",data[z]);
                        printf(" | ");
                    }

                    }
                getch();
            }
        }
    }
}

}while(pil!='4');
}

```

Pengembangan :

1. Buatlah program untuk simulasi antrian registrasi Unimal. Yang tercatat adalah nim mahasiswa.
2. Kembangkan program tersebut sehingga bisa menyimpan keterangan nim, nama, dan prodi mahasiswa tersebut.
3. Tambahkan fitur untuk mengetahui urutan antrian dari nim tertentu.

Soal :

Buat pengembangan program dari latihan di atas dengan beberapa fitur :

1. Bisa melihat urutan untuk prodi tertentu ataupun secara keseluruhan.
2. Bisa POP melompati urutan. Jadi misalkan urutan nomor 2 keluar (POP) lebih dulu daripada urutan pertama.