

# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **LAPORAN AWAL PRAKTIKUM**

**Pertemuan ke-13**

**Queue**



**Disusun Oleh:**

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

**TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PAMULANG**

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566  
Tangerang Selatan - Banten

## A. RANGKUMAN MATERI

Queue atau Antrian merupakan kumpulan elemen dengan penyisipan dan penghapusan elemen yang dilakukan dari sisi/gerbang yang berbeda. Penyisipan dilakukan dari gerbang belakang dan penghapusan dilakukan dari gerbang depan. Hal ini menunjukkan bahwa untuk Queue mempunyai dua gerbang yaitu gerbang depan dan gerbang belakang. Dengan demikian dapat dilihat bahwa Queue mempunyai sifat FIFO (first In First Out), yaitu elemen yang pertama masuk akan keluar pertama juga. Queue dapat direpresentasikan dengan menggunakan Array atau Linked List.

## B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Queue!

**Jawab :**

Queue atau Antrian merupakan kumpulan elemen dengan penyisipan dan penghapusan elemen yang dilakukan dari sisi/gerbang yang berbeda.

2. Tuliskan Deklarasi sintaks Queue!

**Jawab :**

Deklarasi Awal :
<pre>#define max 7; int data[max]; int head = -1, tail = -1;</pre>
IsEmpty :
<pre>bool IsEmpty() {     if (head == -1 &amp;&amp; tail == -1)         return true;     else         return false; }</pre>
IsFull :
<pre>bool IsFull() {     if (tail == max-1)         return true;     else         return false; }</pre>
Enqueue :
<pre>void Enqueue() {     if (IsFull()) {         cout &lt;&lt; "Antrian penuh!" &lt;&lt; endl;     } else {</pre>

<pre> if (IsEmpty()) {     head = tail = 0;     cout &lt;&lt; "Masukkan data: ";     cin &gt;&gt; data[head]; } else {     tail++;     cout &lt;&lt; "Masukkan data: ";     cin &gt;&gt; data[tail]; } } } </pre>
<p>Dequeue :</p> <pre> void Dequeue() {     if (IsEmpty()) {         cout &lt;&lt; "Antrian kosong!" &lt;&lt; endl;     } else {         for (int i = 0; i &lt; tail; i++) {             data[i] = data[i + 1];         }         tail--;     } } </pre>
<p>Clear :</p> <pre> void Clear() {     if (IsEmpty()) {         cout &lt;&lt; "Antrian kosong!" &lt;&lt; endl;     } else {         tail = -1;         cout &lt;&lt; "Antrian berhasil dikosongkan!" &lt;&lt; endl;     } } </pre>
<p>Cetak :</p> <pre> void Cetak() {     if (IsEmpty()) {         cout &lt;&lt; "Antrian kosong!" &lt;&lt; endl;     } else {         cout &lt;&lt; "Isi antrian: ";         for (int i = 0; i &lt;= tail; i++) {             cout &lt;&lt; data[i] &lt;&lt; " ";         }         cout &lt;&lt; endl;     } } </pre>

3. Sebutkan dan Jelaskan operasi-operasi pada Queue!

**Jawab :**

- **Create()** Untuk menciptakan dan menginisialisasi Queue Dengan cara membuat Head dan Tail = -1.
- **IsEmpty()** Untuk memeriksa apakah Antrian sudah penuh atau belum Dengan cara memeriksa nilai Tail, jika Tail = -1 maka empty Kita tidak memeriksa Head, karena Head adalah tanda untuk kepala antrian (elemen pertama dalam antrian) yang tidak akan berubah-ubah Pergerakan pada

Antrian terjadi dengan penambahan elemen Antrian kebelakang, yaitu menggunakan nilai Tail.

- **IsFull()** Untuk mengecek apakah Antrian sudah penuh atau belum Dengan cara mengecek nilai Tail, jika  $\text{Tail} \geq \text{MAX}-1$  (karena  $\text{MAX}-1$  adalah batas elemen array pada C) berarti sudah penuh.
- **Enqueue()** Untuk menambahkan elemen ke dalam Antrian, penambahan elemen selalu ditambahkan di elemen paling belakang Penambahan elemen selalu menggerakkan variabel Tail dengan cara increment counter Tail terlebih dahulu.
- **Dequeue()** Digunakan untuk menghapus elemen terdepan/pertama (head) dari Antrian Dengan cara menggeser semua elemen antrian kedepan dan mengurangi Tail dgn 1 Penggeseran dilakukan dengan menggunakan looping.
- **Clear()** Untuk menghapus elemen-elemen Antrian dengan cara membuat Tail dan Head = -1. Penghapusan elemen-elemen Antrian sebenarnya tidak menghapus arraynya, namun hanya mengeset indeks pengaksesan-nya ke nilai -1 sehingga elemen- elemen Antrian tidak lagi terbaca.
- **Cetak()** Untuk menampilkan nilai-nilai elemen Antrian Menggunakan looping dari head s/d tail.

4. Jelaskan Aplikasi-Aplikasi Queue dalam dunia nyata!

**Jawab :**

Antrian pada penjualan tiket kereta api, dimana orang yang pertama datang adalah orang yang pertama kali dilayani untuk membeli tiket. Jika ada orang baru yang datang akan membeli tiket, maka posisinya berada pada urutan paling belakang dalam antrian tersebut. Orang yang berada pada posisi terakhir dalam antrian adalah yang terakhir kali dapat dilayani dan memperoleh tiket kereta api (kalau kurang beruntung, maka akan kehabisan tiket).

Contoh lain adalah nasabah yang antri di teller bank, paket data yang menunggu untuk di transmisikan lewat internet, antrian printer dimana terdapat antrian print job yang menunggu giliran untuk menggunakan printer dan sebagainya.

Contoh aplikasi Queue dalam dunia nyata :

- Aplikasi antrian di jalan Tol.
- Aplikasi antrian saat mengantri di loket
- Aplikasi antraian reservasi tiket kereta api, dll

Semua itu menggunakan aturan FIFO (First In, First Out).

## C. TUGAS PRAKTIKUM

- Lat13\_1

```
#include <iostream>
using namespace std;

typedef struct node *simpul;
struct node
{
    char Isi;
    simpul Next;
};

void Sisip_Belakang(simpul &L, char elemen);
void Hapus_Depan(simpul &L);
void Cetak(simpul L);

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    char huruf;
    simpul L = NULL;
    int i;

    cout << "== Operasi pada Single Linked List ==" << endl;

    cout << "\nPenyisipan Simpul\n\n";
    for (i = 1; i <= 3; i++)
    {
        cout << "Masukkan huruf : ";
        cin >> huruf;
        Sisip_Belakang(L, huruf);
    }
    Cetak(L);

    cout << "\nSetelah Hapus Simpul\n";
    Hapus_Depan(L);
    Cetak(L);

    cout << "\nSetelah Hapus Simpul\n";
    Hapus_Depan(L);
    Cetak(L);

    cout << "\nSetelah Hapus Simpul\n";
    Hapus_Depan(L);
    Cetak(L);

    cout << "\nPenyisipan Simpul\n\n";
    for (i = 1; i <= 3; i++)
    {
        cout << "Masukkan huruf : ";
        cin >> huruf;
        Sisip_Belakang(L, huruf);
    }
    Cetak(L);

    cout << "\nSetelah Hapus Simpul\n";
    Hapus_Depan(L);
```

```

Cetak(L);

cout << "\nSetelah Hapus Simpul\n";
Hapus_Depan(L);
Cetak(L);

return 0;
}

void Sisip_Belakang(simpul &L, char elemen)
{
    simpul baru = new node;
    baru->Isi = elemen;
    baru->Next = NULL;

    if (L == NULL)
    {
        L = baru;
    }
    else
    {
        simpul bantu = L;
        while (bantu->Next != NULL)
        {
            bantu = bantu->Next;
        }
        bantu->Next = baru;
    }
}

void Cetak(simpul L)
{
    simpul bantu = L;
    if (L == NULL)
    {
        cout << "Linked List kosong....." << endl;
    }
    else
    {
        cout << "\nIsi Linked List : ";
        while (bantu != NULL)
        {
            cout << bantu->Isi << "->";
            bantu = bantu->Next;
        }
    }
}

void Hapus_Depan(simpul &L)
{
    if (L == NULL)
    {
        cout << "Linked List kosong....." << endl;
    }
    else
    {
        simpul Hapus = L;
        L = L->Next;
        Hapus->Next = NULL;
        delete Hapus;
    }
}

```

```
C:\xampp\htdocs\project\teknik-informatika-s1\semester-04\praktikum-struktur-data\pert
Masukkan huruf : a
Masukkan huruf : b
Masukkan huruf : c

Isi Linked List : a->b->c->
Setelah Hapus Simpul

Isi Linked List : b->c->
Setelah Hapus Simpul

Isi Linked List : c->
Setelah Hapus Simpul
Linked List kosong.....

Penyisipan Simpul

Masukkan huruf : d
Masukkan huruf : e
Masukkan huruf : f

Isi Linked List : d->e->f->
Setelah Hapus Simpul

Isi Linked List : e->f->
Setelah Hapus Simpul

Isi Linked List : f->
-----
Process exited after 9.761 seconds with return value 0
Press any key to continue . . .
```



# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **LAPORAN AKHIR PRAKTIKUM**

**Pertemuan ke-13**

**Queue**



**Disusun Oleh:**

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

**TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PAMULANG**

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566  
Tangerang Selatan - Banten

## A. TUGAS AKHIR

1. Buatlah program lat13\_1.cpp diatas, dengan menggunakan system menu!

**Jawab :**

```
#include <iostream>
using namespace std;

typedef struct node *simpul;
struct node
{
    char Isi;
    simpul Next;
};

void Sisip_Belakang(simpul &L, char elemen);
void Hapus_Depan(simpul &L);
void Cetak(simpul L);
void Menu(simpul &L);

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    simpul L = NULL;

    cout << "== Operasi pada Single Linked List ==" << endl;

    Menu(L);

    return 0;
}

void Menu(simpul &L)
{
    int pilihan;
    char huruf;

    while (true)
    {
        cout << "\nMenu Pilihan:" << endl;
        cout << "1. Sisipkan huruf" << endl;
        cout << "2. Hapus huruf dari depan" << endl;
        cout << "3. Cetak linked list" << endl;
        cout << "4. Keluar" << endl;
        cout << "Pilihan Anda: ";
        cin >> pilihan;

        switch (pilihan)
        {
            case 1:
                cout << "Masukkan huruf: ";
                cin >> huruf;
                Sisip_Belakang(L, huruf);
                break;
            case 2:
                Hapus_Depan(L);
                break;
            case 3:
```

```

    Cetak(L);
    break;
case 4:
    return;
default:
    cout << "Pilihan tidak valid. Silakan coba lagi." << endl;
    break;
}
}
}

void Sisip_Belakang(simpul &L, char elemen)
{
    simpul baru = new node;
    baru->Isi = elemen;
    baru->Next = NULL;

    if (L == NULL)
    {
        L = baru;
    }
    else
    {
        simpul bantu = L;
        while (bantu->Next != NULL)
        {
            bantu = bantu->Next;
        }
        bantu->Next = baru;
    }
}

void Cetak(simpul L)
{
    simpul bantu = L;
    if (L == NULL)
    {
        cout << "Linked List kosong....." << endl;
    }
    else
    {
        cout << "\nIsi Linked List : ";
        while (bantu != NULL)
        {
            cout << bantu->Isi << "->";
            bantu = bantu->Next;
        }
    }
}

void Hapus_Depan(simpul &L)
{
    if (L == NULL)
    {
        cout << "Linked List kosong....." << endl;
    }
    else
    {
        simpul Hapus = L;
        L = L->Next;
        Hapus->Next = NULL;
        delete Hapus;
    }
}

```

```

}

Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

== Operasi pada Single Linked List ==

Menu Pilihan:
1. Sisipkan huruf
2. Hapus huruf dari depan
3. Cetak linked list
4. Keluar
Pilihan Anda: 1
Masukkan huruf: a

Menu Pilihan:
1. Sisipkan huruf
2. Hapus huruf dari depan
3. Cetak linked list
4. Keluar
Pilihan Anda: 1
Masukkan huruf: b

Menu Pilihan:
1. Sisipkan huruf
2. Hapus huruf dari depan
3. Cetak linked list
4. Keluar
Pilihan Anda: 3

Isi Linked List : a->b->
Menu Pilihan:
1. Sisipkan huruf
2. Hapus huruf dari depan
3. Cetak linked list
4. Keluar
Pilihan Anda: 4

-----
Process exited after 20.8 seconds with return value 0
Press any key to continue . . .

```

## **2. KESIMPULAN**

Queue mempunyai dua gerbang yaitu gerbang depan dan gerbang belakang. Dengan demikian dapat dilihat bahwa Queue mempunyai sifat FIFO (first In First Out), yaitu elemen yang pertama masuk akan keluar pertama juga. Queue dapat direpresentasikan dengan menggunakan Array atau Linked List..