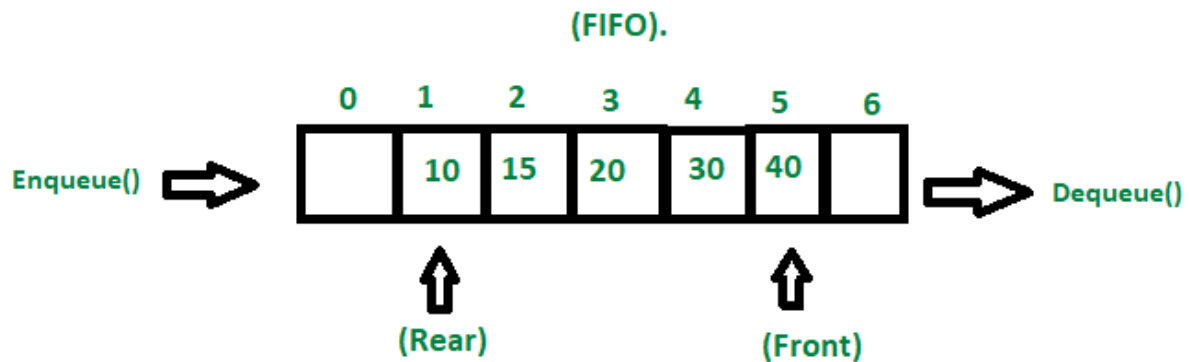


## BAB 6. Queue

### 6.1 Definisi Queue

*Queue* didefinisikan sebagai struktur data linier yang terbuka di kedua ujungnya dan operasi dilakukan dalam urutan First In First Out (FIFO).



sumber: <https://media.geeksforgeeks.org/wp-content/uploads/20220805131014/fifo.png>

Operasi Standar pada Struktur Data Queue:

Enqueue : Memasukkan data ke dalam antrian

Dequeue : Mengeluarkan data terdepan pada antrian

isEmpty : Memeriksa apakah antrian kosong

isFull : Memeriksa apakah antrian penuh

Contoh program 1 :

```

1  #include <iostream>
2  #define maks 20 //maksimum data pada queue
3  using namespace std;
4
5  //deklarasi struct queue
6  struct Queue{
7      int front, rear, data[maks];
8  }Q;
9
10 //cek apakah queue penuh
11 bool isFull() {
12     return Q.rear == maks;
13 }
14
15 //cek apakah queue kosong
16 bool isEmpty() {
17     return Q.rear == 0;
18 }
19
20 //menampilkan queue
21 void printQueue() {
22     if (isEmpty()) {
23         cout<<"Antrian Kosong"<<endl;
24     } else {
25         cout<<"queue : ";
26         for (int i = Q.front; i < Q.rear; i++)
27         {
28             cout<<Q.data[i]<<((Q.rear-1==i) ? "" : ",");
29         }
30         cout<<endl;
31     }
32 }

```

```

34 //enqueue
35 void enqueue() {
36     if(isFull())
37     {
38         cout<<"Queue Penuh"<<endl;
39     }
40     else{
41         int data;
42         cout<<"Masukkan Data : ";
43         cin>>data;
44         Q.data[Q.rear] = data;
45         //menempatkan tail pada elemen data terakhir yang ditambahkan
46         Q.rear++;
47         cout<<"Data Ditambahkan\n";
48         printQueue();
49     }
50 }
51
52 //dequeue
53 void dequeue() {
54     if(isEmpty())
55     {
56         cout<<"Queue Kosong"<<endl;
57     }
58     else{
59         cout<<"Mengeluarkan Data | "<<Q.data[Q.front]<<" |\n";
60         //mengubah queue data ke head
61         for(int i = Q.front; i < Q.rear; i++){
62             Q.data[i] = Q.data[i+1];
63         }
64         Q.rear--;
65         printQueue();
66     }
67 }
68

```

```

68
69
70 int main() {
71     int pilih;
72     do{
73         //Tampilan menu
74         cout << "-----\n"
75         << " Menu Pilihan\n"
76         << "-----\n"
77         << " [1] Enqueue \n"
78         << " [2] Dequeue\n"
79         << " [3] Keluar \n\n"
80         << "-----\n"
81         << "Masukkan pilihan : "; cin>>pilih;
82         switch (pilih){
83             case 1:
84                 enqueue();
85                 break;
86             case 2:
87                 dequeue();
88                 break;
89             default:
90                 cout << "Pilihan tidak tersedia";
91                 break;
92         }
93     } while (pilih !=3);
94     return 0;
95 }
96

```

## Output Program :

```
"D:\KULIAH\Aslab\Str" x + - □ X
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Keluar

-----
Masukkan pilihan : 1
Masukkan Data : 10
Data Ditambahkan
queue : 10
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Keluar

-----
Masukkan pilihan : 1
Masukkan Data : 20
Data Ditambahkan
queue : 10,20
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Keluar

-----
Masukkan pilihan : 1
Masukkan Data : 30
Data Ditambahkan
queue : 10,20,30
-----
Menu Pilihan
-----

"D:\KULIAH\Aslab\Str" x + - □ X
Masukkan Data : 30
Data Ditambahkan
queue : 10,20,30
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Keluar

-----
Masukkan pilihan : 2
Mengeluarkan Data | 10 |
queue : 20,30
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Keluar

-----
Masukkan pilihan : 2
Mengeluarkan Data | 20 |
queue : 30
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Keluar

-----
Masukkan pilihan : 3
Pilihan tidak tersedia
Process returned 0 (0x0)   execution time :
102.404 s
Press any key to continue.
```

## Contoh program 2 :

```
1  #include <iostream>
2
3  using namespace std;
4
5  struct Node {
6      int data;
7      Node* next;
8  };
9
10 struct Queue {
11     Node* front;
12     Node* rear;
13 };
14
15 // Inisialisasi queue
16 void initQueue(Queue* q) {
17     q->front = q->rear = NULL;
18 }
19
20 // Cek apakah queue kosong
21 bool isEmpty(Queue* q) {
22     return (q->front == NULL && q->rear == NULL);
23 }
24
25 // Enqueue data
26 void enqueue(Queue* q, int data) {
27     Node* newNode = new Node;
28     newNode->data = data;
29     newNode->next = NULL;
30     if (isEmpty(q)) {
31         q->front = q->rear = newNode;
32     }
33     else {
34         q->rear->next = newNode;
35         q->rear = newNode;
36     }
37 }
```

```

39 // Dequeue data
40 void dequeue(Queue* q) {
41     if (isEmpty(q)) {
42         cout << "Queue kosong" << endl;
43     }
44     else {
45         Node* temp = q->front;
46         q->front = q->front->next;
47         if (q->front == NULL) {
48             q->rear = NULL;
49         }
50         cout << "Mengeluarkan data: " << temp->data << endl;
51         delete(temp);
52     }
53 }
54
55 // Tampilkan isi queue
56 void display(Queue* q) {
57     if (isEmpty(q)) {
58         cout << "Queue kosong" << endl;
59     }
60     else {
61         Node* temp = q->front;
62         while (temp != NULL) {
63             cout << temp->data << " ";
64             temp = temp->next;
65         }
66         cout << endl;
67     }
68 }
69
70 int main() {
71     Queue q;
72     initQueue(&q);
73
74     // Menu
75     int pilih, data;

```

```

74 // Menu
75 int pilih, data;
76 do {
77     cout << "-----" << endl;
78     cout << " Menu Pilihan" << endl;
79     cout << "-----" << endl;
80     cout << " [1] Enqueue" << endl;
81     cout << " [2] Dequeue" << endl;
82     cout << " [3] Tampilkan" << endl;
83     cout << " [4] Keluar" << endl;
84     cout << "-----" << endl;
85     cout << "Masukkan pilihan: ";
86     cin >> pilih;
87
88     switch (pilih) {
89         case 1:
90             cout << "Masukkan data: ";
91             cin >> data;
92             enqueue(&q, data);
93             break;
94         case 2:
95             dequeue(&q);
96             break;
97         case 3:
98             display(&q);
99             break;
100        case 4:
101            cout << "Keluar" << endl;
102            break;
103        default:
104            cout << "Pilihan tidak tersedia" << endl;
105        }
106    } while (pilih != 4);
107
108    return 0;
109 }
110

```



Output program :

```
-----
[1] Enqueue
[2] Dequeue
[3] Tampilkan
[4] Keluar
-----
Masukkan pilihan: 1
Masukkan data: 1
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Tampilkan
[4] Keluar
-----
Masukkan pilihan: 1
Masukkan data: 2
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Tampilkan
[4] Keluar
-----
Masukkan pilihan: 1
Masukkan data: 3
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Tampilkan
[4] Keluar
-----
Masukkan pilihan: 3
1 2 3
-----
-----
Masukkan pilihan: 3
1 2 3
-----
-----
Masukkan pilihan: 3
1 2 3
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Tampilkan
[4] Keluar
-----
Masukkan pilihan: 2
Mengeluarkan data: 1
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Tampilkan
[4] Keluar
-----
Masukkan pilihan: 3
2 3
-----
Menu Pilihan
-----
[1] Enqueue
[2] Dequeue
[3] Tampilkan
[4] Keluar
-----
Masukkan pilihan: 4
Keluar
```

**Latihan :**

Buatlah sebuah program yang menggunakan queue untuk mengecek apakah suatu string adalah palindrome atau tidak.

**Contoh**

Input:

bbaabb

output:

Queue adalah palindrome