

Modul 8 : ADT Queue

8.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum = **170 menit**, dengan rincian sebagai berikut (misalkan):

- 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan tugas / Studi Kasus
- 50 menit **Pengayaan**

8.2 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Mampu membuat queue menggunakan array
2. Mampu membuat queue menggunakan linked list.....

8.3 Alat & Bahan

1. Komputer
2. Java IDE

8.4 Dasar Teori

Queue atau antrian adalah kumpulan data yang bisa dilakukan penambahan (penyisipan) data pada satu sisi dan penghapusan data pada sisi/ujung yang lain.



Dengan memperhatikan ilustrasi diatas maka kita bisa melihat bahwa queue merupakan suatu list yang mempunyai karakteristik “masuk pertama keluar pertama” (first in first out – FIFO). Struktur data ini banyak dipakai dalam informatika misalnya untuk merepresentasi :

1. Antrian job dalam sistem operasi
2. Antrian dalam dunia nyata

Operasi dasar pada queue antara lain **enqueue** dan **dequeue**. **Enqueue** adalah operasi menambah/menyisipkan item pada akhir queue. Pada proses enqueue item ditambahkan di belakang item terakhir dalam queue. **Dequeue** adalah proses mengambil item dari queue. Ketika proses dequeue item dalam queue akan berkurang yaitu item paling depan dalam queue atau item yang pertama kali ditambahkan kedalam queue. Queue bisa dibuat dengan Array atau Linked List. Tabel dibawah memberikan ilustrasi operasi pada queue yang dibuat dengan array.

sebelum	operasi	sesudah
[]	queue("A")	["A"]
["A"]	queue("B")	["A","B"]
["A","B"]	queue("C")	["A","B","C"]
["A","B","C"]	dequeue()	["B","C"]
["A","B"]	queue("D")	["A","B","D"]
["A","B","D"]	dequeue()	["B","D"]

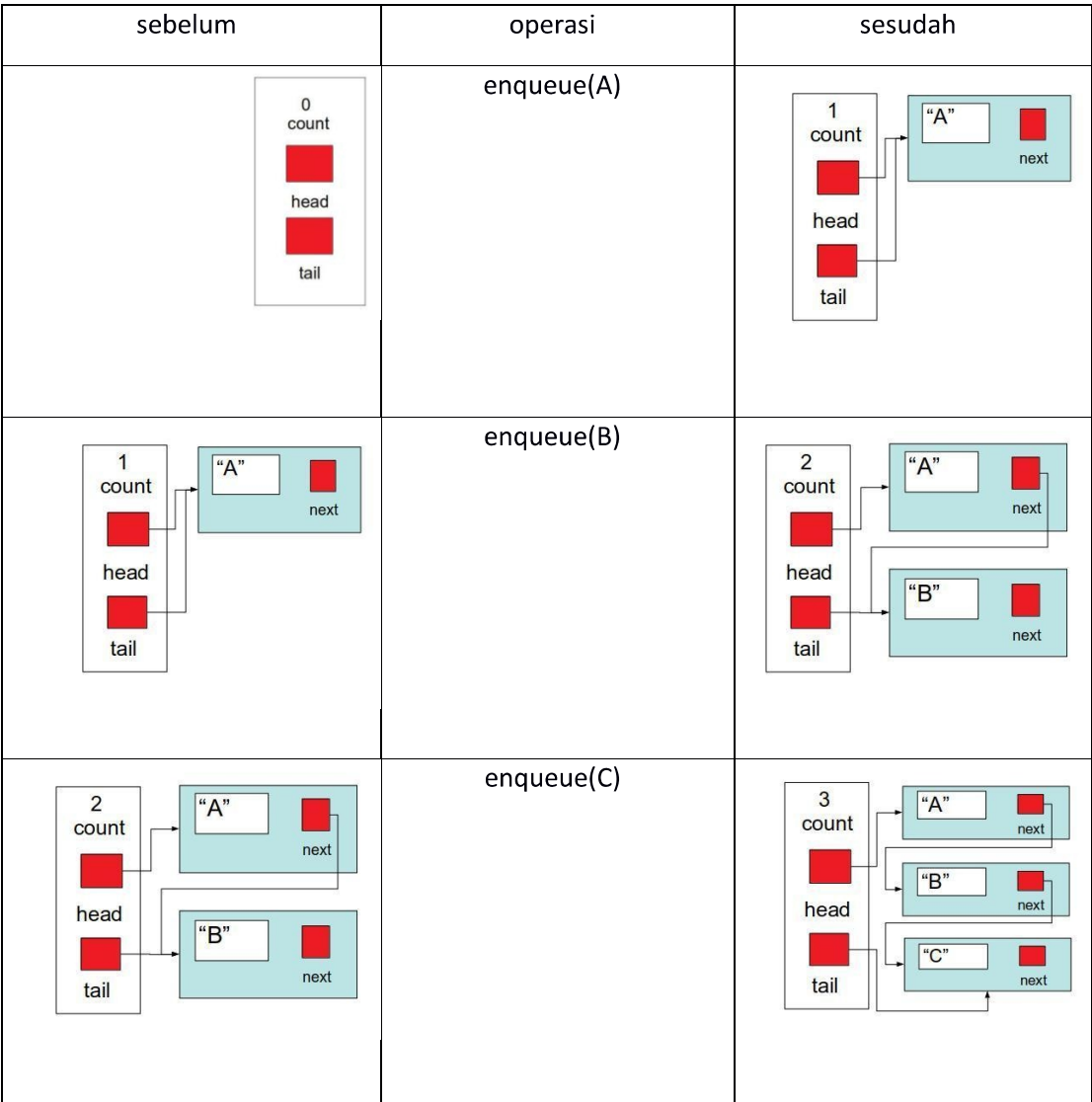
Dalam implementasi queue menggunakan array ini disimpan variabel **rear** yang menunjukkan index atau nomor urut terakhir item dalam queue. Operasi **enqueue** menambahkan item ke dalam array pada index/urutan **rear+1** atau **setelah item terakhir** dalam array. Operasi **dequeue** akan mengambil item dari dalam array pada index 0 atau **item pertama/paling depan** dalam array. Kemudian dilakukan pergeseran isi array kekiri sebanyak satu dan mengurangi rear sebanyak satu(rear-1).

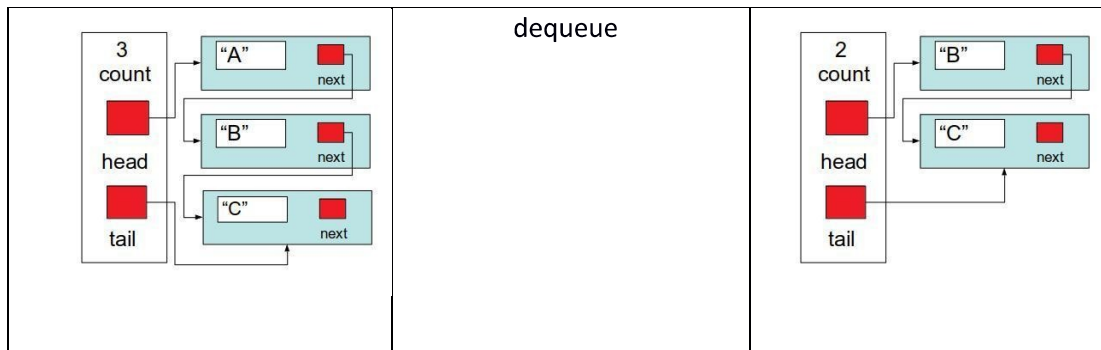
Pada tabel dibawah diilustrasikan operasi queue menggunakan array. Perbedaan dengan queue diatas adalah implementasi proses **enqueue** akan menggeser item dalam array kekanan sebanyak satu dan menambahkan/menyisipkan item pada **array index ke 0**. Proses **dequeue** akan mengambil item pada **array index terakhir atau rear**.

sebelum	operasi	sesudah
[]	queue("A")	["A"]
["A"]	queue("B")	["B","A"]
["B","A"]	queue("C")	["C","B","A"]
["C","B","A"]	dequeue()	["C","B"]

["C","B"]	queue("D")	["D","C","B"]
["D","C","B"]	dequeue()	["D","C"]

Gambar dibawah memberikan ilustrasi queue menggunakan single linked list. Head merupakan pointer yang menunjuk pada item paling atas yang merupakan item yang pertama dimasukkan dalam queue, item paling depan dalam antrian. Tail merupakan pointer yang selalu menunjuk pada item yang terakhir ditambahkan dalam queue, item paling belakang dalam antrian. Dalam ilustrasi ini operasi **enqueue** pada queue menggunakan operasi **add last** pada single linked list dan operasi **dequeue** menggunakan operasi **remove first**. Jika operasi **enqueue** menggunakan operasi **add first** maka operasi **dequeue** akan menggunakan **remove last**.





8.5 Prosedur Praktikum

1. Buatlah file percobaan queue_array.java berikut

```

1  import java.util.Scanner;
2
3  Public class queue_array {
4  Scanner masuk = new Scanner(System.in);
5
6  int choice, i;
7  char item;
8  char arr_stack[MAX_SIZE];
9  int count = 0;
10 int keluar = 0;
11 int front,rear=0;
12
13 Public void enqueue(char item)
14 {
15     if (rear == MAX_SIZE)
16     {
17         system.out.print ( "\n# Queue Penuh");
18     }
19     else
20     {
21         system.out.print ( "\n# Queue No urut/index : "+ rear+ ", Queue :"+ item);
22         .....
23     }
24 }
25 Public void dequeue()
26 {
27     if (rear == 0)
28     system.out.print ( "\n## Queue kosong");
29     else
30     {
31         .....
32         .....
33         .....
34         .....
35         .....
36         .....

```

```

37     .....
38     }
39 }
40
41 Public void printAll()
42 {
43     system.out.print ( "\n## Queue Size : " + rear);
44     for (i = 0; i < rear; i++)
45         system.out.print ( "\n## No Urut/index : " + i + ", Value : " + arr_stack[i]);
46 }
47
48 Public void menu()
49 {
50     system.out.print ( "\nMasukkan operasi yang akan dilakukan (1:enqueue, 2:dequeue,
51     3:print) : ");
52     choice= masuk.nextInt();
53     switch (choice)
54     {
55     case 1:
56     {
57         system.out.print ( "\nMasukkan huruf yang akan di-enqueue : ");
58         item= masuk.nextLine();
59         enqueue(item);
60     }
61     case 2:
62     {
63         dequeue();
64     }
65     case 3:
66     {
67         printAll();
68     }
69     default:
70     {
71         system.out.print ( "\n1:enqueue, 2:dequeue, 3:print\n");
72         keluar = 1;
73         break;
74     }
75 }
76
77 Public static void main()
78 {
79     do
80     {
81         menu();
82     } while (keluar == 0);
83 }

```

2. Tambahkan kode berikut pada baris 22

22	arr_stack[rear++] = item;
----	---------------------------

3. Tambahkan kode berikut pada baris 32 sampai 38

```

32     System.out.print ( "\n##Dequeue Value :" +
33     arr_stack[0]);
34     for (i=1;i<=rear;i++)
35     {
36         char temp=arr_stack[i];
37         arr_stack[i-1]=temp;
38     }
    rear--;

```

4. Jalankan program kemudian

- pilih menu enqueue dan masukkan “A”
- pilih menu enqueue dan masukkan “B”
- pilih menu enqueue dan masukkan “C”
- pilih menu print
- pilih menu dequeue
- pilih menu print
- pilih menu enqueue dan masukkan “D”
- pilih menu print

8.6 Hasil Percobaan

Tuliskan hasil dari percobaan diatas.

8.7 Analisis Hasil

Tuliskan Analisis hasil dari percobaan diatas.

8.8 Kesimpulan

Tuliskan kesimpulan dari percobaan diatas.

8.9 Latihan

Buatlah program queue menggunakan array dengan ketentuan operasi enqueue akan selalu menambahkan item kedalam array pada index ke 0.

8.10 Tugas

Buatlah program queue menggunakan double linked list dengan ketentuan operasi enqueue akan menambahkan item pada node head(add first) dan operasi dequeue akan mengambil item dari node pada tail(remove last).

8.11 DAFTAR PUSTAKA

- Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, “Data Structures and Algorithms Using Java 6 edition”, Wiley, USA, 2014.
- John R. Hubbard, “Scaum’s Outline of Data Structures With Java second Edition”, McGraw-Hill, New york, 2007.
- Robert Lafore, “Data Structures and Algorithm in Java second Edition”, Sams Publishing, Indiana, 2003