

LAPORAN PRAKTIKUM
STRUKTUR DATA
PERTEMUAN KE – 6



Disusun Oleh :

NAMA : TARISA DWI SEPTIA
NIM : 205410126
JURUSAN : TEKNIK INFORMATIKA
JENJANG : S1

Sekolah Tinggi Management Informatika Komputer

AKAKOM

YOGYAKARTA

2020

MODUL 6

PEMANFAATAN ARRAY/LARIK SEBAGAI STACK (TUMPUKAN) DAN QUEUE (ANTRIAN)

A. Tujuan

- Mahasiswa dapat mengimplementasikan tumpukan dan antrian untuk berbagai keperluan dengan menggunakan bahasa pemrograman Java

B. Praktik

a. Praktik 1

Tuliskan program 7.1 menggunakan TextPad. Tambahkan pada bagian program utama operasi “push” berturut-turut :

```
push (tumpukan, "Buku A");
push (tumpukan, "Buku B");
push (tumpukan, "Buku C");
push (tumpukan, "Buku D");
bacaTumpukan(tumpukan);
```

Sekarang jalankan program diatas, dan amati yang terjadi. Buku apa sajakah yang dapat di push ke dalam tumpukan? Apakah buku A, B, C, D, keempatnya dapat dipush ke dalam tumpukan? Amatilah posisi “atas”. Menurut anda, setelah dipush nya keempat buku di ABCD, berada di posisi manakah “atas”? Catatlah dalam laporan anda.

```
public class P1{
    public static int N = 5;
    public static int atas = -1;
    public static void push (String tumpukan[], String data){
        if (atas == N-1){ //jika tumpukan penuh
            System.out.println("maap, tumpukan penuh, PUSH "
+ data+ " tidak dapat dilakukan");
        }else{
            //System.out.println(atas);
            //jika tumpukan tidak penuh
            atas = atas + 1;
            tumpukan[atas] = data;
            System.out.println("PUSH " + data + " berhasil..");
        }
    }
    public static String POP (String tumpukan[]){
        String hasil;
        if (atas < 0){ //jika tumpukan kosong
            hasil = "TUMPUKAN KOSONG, POP GAGAL
DILAKUKAN";
        }else{ //jika tumpukan tidak kosong
            hasil = tumpukan[atas];
            atas = atas - 1;
        }
        return (hasil);
    }
    public static void lihatTumpukan(String tumpukan[]){
        System.out.println("");
        System.out.println("--TUMPUKAN:-");
    }
}
```

```

        for (int i=atas; i>=0; i--){
            System.out.println("index ke "+i+ ": "+tumpukan[i]);

        }
        System.out.println("--akhir tumpukan-");
        System.out.println("");
    }
    public static void main (String[] args){
        String tumpukan[] = new String[10];
        push(tumpukan, "Buku A");
        push(tumpukan, "Buku B");
        push(tumpukan, "Buku C");
        push(tumpukan, "Buku D");
        lihatTumpukan(tumpukan);
    }
}

```

Output :

```

PUSH Buku A berhasil..
PUSH Buku B berhasil..
PUSH Buku C berhasil..
PUSH Buku D berhasil..

--TUMPUKAN:-
index ke 3: Buku D
index ke 2: Buku C
index ke 1: Buku B
index ke 0: Buku A
--akhir tumpukan-

Press any key to continue . . . _

```

Ke empat buku yaitu buku A, B, C dan buku D dapat di push semua kedalam tumpukan dan variable atas menempatai array ke [3] karena atas dimulai dari [0] sehingga pada push buku ke empat variable atas menempati posisi array ke [3]

b. Praktik 2

Sekarang, lakukan lagi proses push untuk sederet data “buku” berikut ini ke dalam tumpukan:

```

push (tumpukan, "Buku E");
push (tumpukan, "Buku F");
push (tumpukan, "Buku G");
bacaTumpukan(tumpukan);

```

Sekarang jalankan program diatas, dan amati yang terjadi. “Buku” apa sajakah yang dapat di push ke dalam tumpukan? Adakah “Buku” yang gagal di push ke dalam tumpukan? Apa sebabnya? Mengapa bisa demikian? Menurut anda bagian manakah dari program di atas yang menyebabkannya? Jelaskanlah dalam laporan anda.

```

public class P2{
    public static int N = 5;
    public static int atas = -1;
    public static void push (String tumpukan[], String data){
        if (atas == N-1){ //jika tumpukan penuh

```

```

        System.out.println("maap, tumpukan penuh, PUSH " + data+
" tidak dapat dilakukan");
    }else{ //jika tumpukan tidak penuh
        atas = atas + 1;
        tumpukan[atas] = data;
        System.out.println("PUSH " + data + " berhasil..");
    }
}

public static String POP (String tumpukan[]){
    String hasil;
    if (atas < 0 ){ //jika tumpukan kosong
        hasil = "TUMPUKAN KOSONG, POP GAGAL
DILAKUKAN";
    }else{ //jika tumpukan tidak kosong
        hasil = tumpukan[atas];
        atas = atas - 1;
    }
    return (hasil);
}

public static void lihatTumpukan(String tumpukan[]){
    System.out.println("");
    System.out.println("--TUMPUKAN:-");
    for (int i=atas; i>=0; i--){
        System.out.println(tumpukan[i]);
    }
    System.out.println("--akhir tumpukan-");
    System.out.println("");
}

public static void main (String[] args){
    String tumpukan[] = new String[10];
    push(tumpukan, "Buku A");
    push(tumpukan, "Buku B");
    push(tumpukan, "Buku C");
    push(tumpukan, "Buku D");
    push(tumpukan, "Buku E");
    push(tumpukan, "Buku F");
    push(tumpukan, "Buku G");
    lihatTumpukan(tumpukan);
}
}

```

Output :

```
PUSH Buku A berhasil..
PUSH Buku B berhasil..
PUSH Buku C berhasil..
PUSH Buku D berhasil..
PUSH Buku E berhasil..
maap, tumpukan penuh, PUSH Buku F tidak dapat dilakukan
maap, tumpukan penuh, PUSH Buku G tidak dapat dilakukan

--TUMPUKAN:-
Buku E
Buku D
Buku C
Buku B
Buku A
--akhir tumpukan-

Press any key to continue . . .
```

Buku yang dapat di push ketumpukan hanya buku A sampai E dikarenakan variable N yang bernilai 5 yang ber arti tumpukan hanya boleh max 5 buku saja sehingga buku F dan G tidak dapat kepush ketumpukan.

c. Praktik 3

Sekarang, lakukan pop dari tumpukan dengan cara menambahkan perintah berikut ke bagian akhir dari program utama .

```
System.out.println("POP: " + pop(tumpukan));
bacaTumpukan(tumpukan);
```

“Buku” apakah yang ter-pop ? Mengapa bisa demikian? Sekarang amatilah kondisi tumpukan saat ini. Adakah yang berkurang? Mengapa hal itu bisa terjadi? Menurut anda, bagian manakah dari program di atas yang menyebabkan itu terjadi? Bagaimanakah juga dengan kondisi “atas” saat ini? Jelaskan pada laporan anda.

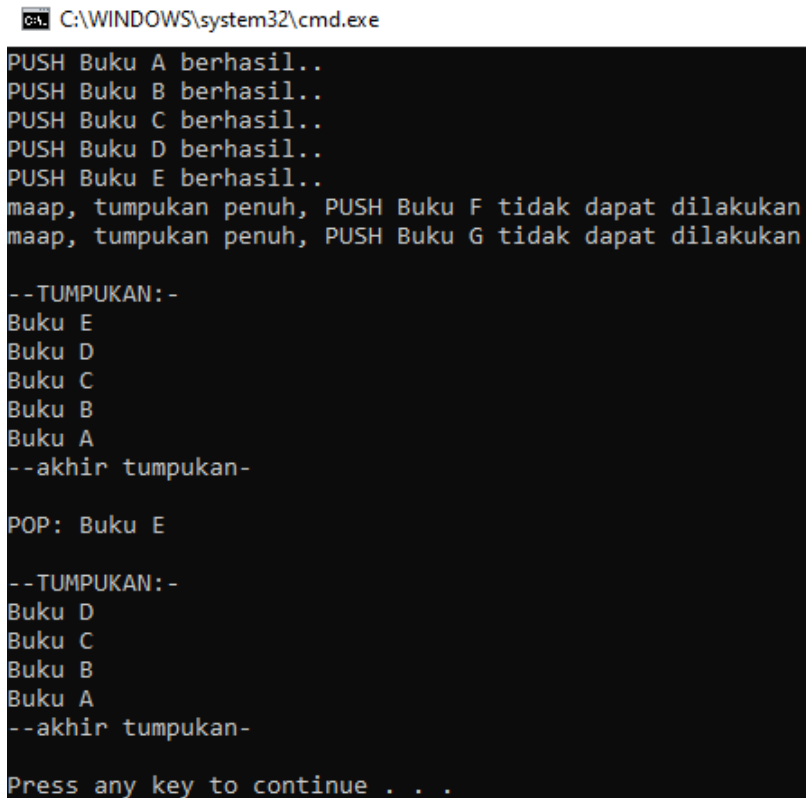
```
public class P3{
    public static int N = 5;
    public static int atas = -1;
    public static void push (String tumpukan[], String data){
        if (atas == N-1){ //jika tumpukan penuh
            System.out.println("maap, tumpukan penuh, PUSH " + data+
" tidak dapat dilakukan");
        }else{ //jika tumpukan tidak penuh
            atas = atas + 1;
            tumpukan[atas] = data;
            System.out.println("PUSH " + data + " berhasil..");
        }
    }
    public static String pop (String tumpukan[]){
        String hasil;
        if (atas < 0 ){ //jika tumpukan kosong
            hasil = "TUMPUKAN KOSONG, POP GAGAL
DILAKUKAN";
        }else{ //jika tumpukan tidak kosong
            hasil = tumpukan[atas];
            atas = atas - 1;
        }
    }
}
```

```

    }
    return (hasil);
}
public static void lihatTumpukan(String tumpukan[]){
    System.out.println("");
    System.out.println("--TUMPUKAN:-");
    for (int i=atas; i>=0; i--){
        System.out.println(tumpukan[i]);
    }
    System.out.println("--akhir tumpukan-");
    System.out.println("");
}
public static void main (String[] args){
    String tumpukan[] = new String[10];
    push(tumpukan, "Buku A");
    push(tumpukan, "Buku B");
    push(tumpukan, "Buku C");
    push(tumpukan, "Buku D");
    push(tumpukan, "Buku E");
    push(tumpukan, "Buku F");
    push(tumpukan, "Buku G");
    lihatTumpukan(tumpukan);
    System.out.println("POP: " + pop(tumpukan));
    lihatTumpukan(tumpukan);
}
}

```

Output :



```

C:\WINDOWS\system32\cmd.exe
PUSH Buku A berhasil..
PUSH Buku B berhasil..
PUSH Buku C berhasil..
PUSH Buku D berhasil..
PUSH Buku E berhasil..
maap, tumpukan penuh, PUSH Buku F tidak dapat dilakukan
maap, tumpukan penuh, PUSH Buku G tidak dapat dilakukan

--TUMPUKAN:-
Buku E
Buku D
Buku C
Buku B
Buku A
--akhir tumpukan-

POP: Buku E

--TUMPUKAN:-
Buku D
Buku C
Buku B
Buku A
--akhir tumpukan-

Press any key to continue . . .

```

Buku yang ter-pop adalah buku E karena buku E berada di posisi teratas, dikarenakan buku E ter-pop maka tumpukan sudah berkurang 1 buku. Program pada **class String pop** yang menyebabkan bisa ter-pop nya buku E. Setelah di pop maka variable atas akan berkurang 1, itu lah yang menyebabkan tumpukan berkurang 1 buku.

d. Praktik 4

Sekarang, lakukan lagi proses pop dari tumpukan sebanyak 2 kali berturut-turut :

```
System.out.println("POP: " + pop(tumpukan);
System.out.println("POP: " + pop(tumpukan);
bacaTumpukan(tumpukan);
```

Catatlah “Buku” apakah saja yang ter-‘pop’ ? Sekarang amatilah kondisi tumpukan saat ini. Adakah perubahan yang terjadi pada kondisi tumpukan saat ini dibanding dengan pada percobaan sebelumnya? Mengapa hal itu bisa terjadi? Bagaimanakah juga dengan kondisi “atas” saat ini? Jelaskan pada laporan anda.

```
public class P4{
    public static int N = 5;
    public static int atas = -1;
    public static void push (String tumpukan[], String data){
        if (atas == N-1){ //jika tumpukan penuh
            System.out.println("maap, tumpukan penuh, PUSH " + data+
" tidak dapat dilakukan");
        }else{ //jika tumpukan tidak penuh
            atas = atas + 1;
            tumpukan[atas] = data;
            System.out.println("PUSH " + data + " berhasil..");
        }
    }
    public static String pop (String tumpukan[]){
        String hasil;
        if (atas < 0 ){ //jika tumpukan kosong
            hasil = "TUMPUKAN KOSONG, POP GAGAL
DILAKUKAN";
        }else{ //jika tumpukan tidak kosong
            hasil = tumpukan[atas];
            atas = atas - 1;
        }
        return (hasil);
    }
    public static void lihatTumpukan(String tumpukan[]){
        System.out.println("");
        System.out.println("--TUMPUKAN:-");
        for (int i=atas; i>=0; i--){
            System.out.println(tumpukan[i]);
        }
        System.out.println("--akhir tumpukan-");
        System.out.println("");
    }
    public static void main (String[] args){
        String tumpukan[] = new String[10];
        push(tumpukan, "Buku A");
```

```

push(tumpukan, "Buku B");
push(tumpukan, "Buku C");
push(tumpukan, "Buku D");
push(tumpukan, "Buku E");
push(tumpukan, "Buku F");
push(tumpukan, "Buku G");
lihatTumpukan(tumpukan);
System.out.println("POP: " + pop(tumpukan));
System.out.println("POP: " + pop(tumpukan));
lihatTumpukan(tumpukan);

```

```

}

```

```

}

```

Output :

```

C:\WINDOWS\system32\cmd.exe
PUSH Buku A berhasil..
PUSH Buku B berhasil..
PUSH Buku C berhasil..
PUSH Buku D berhasil..
PUSH Buku E berhasil..
maap, tumpukan penuh, PUSH Buku F tidak dapat dilakukan
maap, tumpukan penuh, PUSH Buku G tidak dapat dilakukan

--TUMPUKAN:-
Buku E
Buku D
Buku C
Buku B
Buku A
--akhir tumpukan-

POP: Buku E
POP: Buku D

--TUMPUKAN:-
Buku C
Buku B
Buku A
--akhir tumpukan-

Press any key to continue . . .

```

Buku yang ter-pop pada program ini bertambah menjadi 2 buku, yaitu buku E dan buku D. Variable atas sekarang juga berkurang 2 menjadi atas[3].

e. Praktik 5

Sekarang, lakukan lagi proses pop dari tumpukan sebanyak 3 kali berturut-turut :

```

System.out.println("POP: " + pop(tumpukan);
System.out.println("POP: " + pop(tumpukan);
System.out.println("POP: " + pop(tumpukan);
bacaTumpukan(tumpukan);

```

Adakah proses pop yang tidak dapat (gagal) dilakukan? Mengapa bisa demikian?
Jelaskan pada laporan anda

```

public class P5{
    public static int N = 5;
    public static int atas = -1;
    public static void push (String tumpukan[], String data){

```


Output :

```
C:\WINDOWS\system32\cmd.exe
PUSH Buku A berhasil..
PUSH Buku B berhasil..
PUSH Buku C berhasil..
PUSH Buku D berhasil..
PUSH Buku E berhasil..
maap, tumpukan penuh, PUSH Buku F tidak dapat dilakukan
maap, tumpukan penuh, PUSH Buku G tidak dapat dilakukan

--TUMPUKAN:-
Buku E
Buku D
Buku C
Buku B
Buku A
--akhir tumpukan-

POP: Buku E
POP: Buku D
POP: Buku C
POP: Buku B
POP: Buku A

--TUMPUKAN:-
--akhir tumpukan-

Press any key to continue . . . _
```

Semua tumpukan bisa ter-pop dikarenakan masih cukup buku yang akan di pop.

f. Praktik 6

Tuliskan program 7.2 menggunakan TextPad. Tambahkan pada bagian program utama operasi “enQueue” berturut-turut :

```
enQueue (antrian, "Mobil A");
enQueue (antrian, "Mobil B");
enQueue (antrian, "Mobil C");
bacaAntrian(antrian);
```

Sekarang jalankan program diatas, dan amati yang terjadi. “Mobil” apa sajakah yang dapat di enQueue ke dalam antrian? Lalu bagaimanakah dengan ‘belakang’? Berada di posisi manakah ‘belakang’ saat ini? Catatlah dalam laporan anda

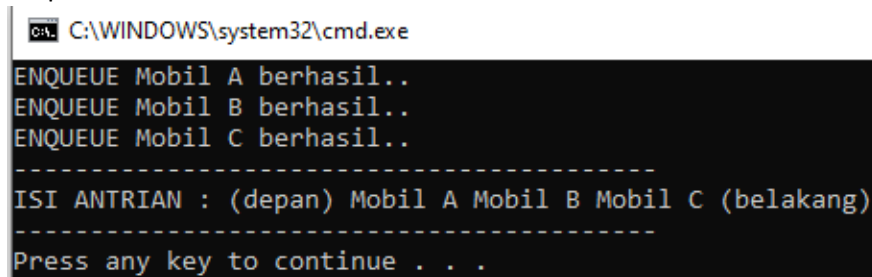
```
public class P6{
    public static int N = 5;
    public static int belakang = -1;
    public static void enQueue (String antrian[], String data){
        if (belakang == N-1){
            //jika antrian penuh
            System.out.println("maap, antrian penuh, ENQUEUE "
+ data + " tidak dapat dilakukan");
        }else{ //jika antrian tidak penuh
            belakang = belakang + 1;
            antrian[belakang] = data;
            System.out.println("ENQUEUE " + data + " berhasil..");
        }
    }
    public static String DEQUEUE (String antrian[]){
        String hasil;
```

```

        if (belakang < 0 ){ //jika antrian kosong
            hasil = "ANTRIAN KOSONG, DEQUEUE GAGAL
DILAKUKAN";
        }else{ //jika antrian tidak kosong
            hasil = antrian[0];
            //----menggeser data kedua dst, maju selangkah ke
            depan
            for (int i=0; i<=belakang-1; i++){
                antrian[i] = antrian[i+1];
            }
            belakang = belakang - 1;
        }
        return (hasil);
    }
    public static void lihatAntrian(String antrian[]){
        System.out.println("-----");
        System.out.print("ISI ANTRIAN : (depan)");
        for (int i=0; i<=belakang; i++){
            System.out.print(" " + antrian[i]);
        }
        System.out.println(" (belakang)");
        System.out.println("-----");
    }
    public static void main (String[] args){
        String antrian[] = new String[5];
        enqueue (antrian, "Mobil A");
        enqueue (antrian, "Mobil B");
        enqueue (antrian, "Mobil C");
        lihatAntrian(antrian);
    }
}

```

Output :



```

C:\WINDOWS\system32\cmd.exe
ENQUEUE Mobil A berhasil..
ENQUEUE Mobil B berhasil..
ENQUEUE Mobil C berhasil..
-----
ISI ANTRIAN : (depan) Mobil A Mobil B Mobil C (belakang)
-----
Press any key to continue . . .

```

Semua mobil dapat di enqueue dikarenakan masih kurang dari batas yang telah ditentukan pada variable N. Posisi belakang sekarang di isi oleh mobil C.

g. Praktik 7

Sekarang, lakukan lagi proses enqueue sederet data "Mobil" berikut ini ke dalam antrian:

```

enqueue (antrian, "Mobil E");
enqueue (antrian, "Mobil F");
enqueue (antrian, "Mobil G");
lihatAntrian(antrian);

```

Sekarang jalankan program diatas, dan amati yang terjadi. Bagaimanakah kondisi antrian saat ini? “Mobil” apa sajakah yang ada dalam antrian? Adakah “Mobil” yang gagal di enqueue ke dalam antrian? Apa sebabnya? Mengapa bisa demikian? Menurut anda bagian manakah dari program di atas yang menyebabkannya? Jelaskanlah dalam laporan anda. Lalu bagaimana dengan posisi “belakang” saat ini? Adakah perbedaan posisi “belakang” saat ini jika dibandingkan dengan posisi “belakang” pada pelaksanaan praktikum ke-6. Catatlah dalam laporan anda.

```

public class P7{
    public static int N = 5;
    public static int belakang = -1;
    public static void enqueue (String antrian[], String data){
        if (belakang == N-1){
            //jika antrian penuh
            System.out.println("maap, antrian penuh, ENQUEUE "
+ data + " tidak dapat dilakukan");
        }else{ //jika antrian tidak penuh
            belakang = belakang + 1;
            antrian[belakang] = data;
            System.out.println("ENQUEUE " + data + " berhasil..");
        }
    }
    public static String dequeue (String antrian[]){
        String hasil;
        if (belakang < 0 ){ //jika antrian kosong
            hasil = "ANTRIAN KOSONG, DEQUEUE GAGAL
DILAKUKAN";
        }else{ //jika antrian tidak kosong
            hasil = antrian[0];
            //----menggeser data kedua dst, maju selangkah ke
depan
            for (int i=0; i<=belakang-1; i++){
                antrian[i] = antrian[i+1];
            }
            belakang = belakang - 1;
        }
        return (hasil);
    }
    public static void lihatAntrian(String antrian[]){
        System.out.println("-----");
        System.out.print("ISI ANTRIAN : (depan)");
        for (int i=0; i<=belakang; i++){
            System.out.print(" " + antrian[i]);
        }
        System.out.println(" (belakang)");
        System.out.println("-----");
    }
    public static void main (String[] args){
        String antrian[] = new String[5];
        enqueue (antrian, "Mobil A");
        enqueue (antrian, "Mobil B");
        enqueue (antrian, "Mobil C");
    }
}

```

```

        enqueue (antrian, "Mobil D");
        enqueue (antrian, "Mobil E");
        enqueue (antrian, "Mobil F");
        lihatAntrian(antrian);
    }
}

```

Output :

```

C:\WINDOWS\system32\cmd.exe
ENQUEUE Mobil A berhasil..
ENQUEUE Mobil B berhasil..
ENQUEUE Mobil C berhasil..
ENQUEUE Mobil D berhasil..
ENQUEUE Mobil E berhasil..
maap, antrian penuh, ENQUEUE Mobil F tidak dapat dilakukan
-----
ISI ANTRIAN : (depan) Mobil A Mobil B Mobil C Mobil D Mobil E (belakang)
-----
Press any key to continue . . .

```

Hanya mobil A sampai mobil E yang berada di antrian dikarenakan mobil yang di enqueue sudah melebihi batas yang telah di tentukan oleh var N. Posisi belakang juga berubah menjadi mobil E dikarenakan antrian terakhir telah di isi oleh mobil E.

h. Praktik 8

Sekarang, lakukan dequeue dari antrian dengan cara menambahkan perintah berikut ke bagian akhir dari program utama.

```

System.out.println("deQueue: " + dequeue(antrian));
lihatAntrian(antrian);

```

“Mobil” apakah yang ter-deQueue ? Mengapa bisa demikian? Sekarang amatilah kondisi antrian saat ini. Adakah perubahan yang terjadi pada kondisi antrian saat ini dibanding dengan pada percobaan sebelumnya? Apakah setiap data “Mobil B,C,D,..dst” masih berada di posisinya yang sama? Mengapa hal itu bisa terjadi? Menurut anda, bagian manakah dari program di atas yang menyebabkan itu terjadi? Bagaimanakah juga dengan kondisi ekor saat ini? Jelaskan pada laporan anda.

```

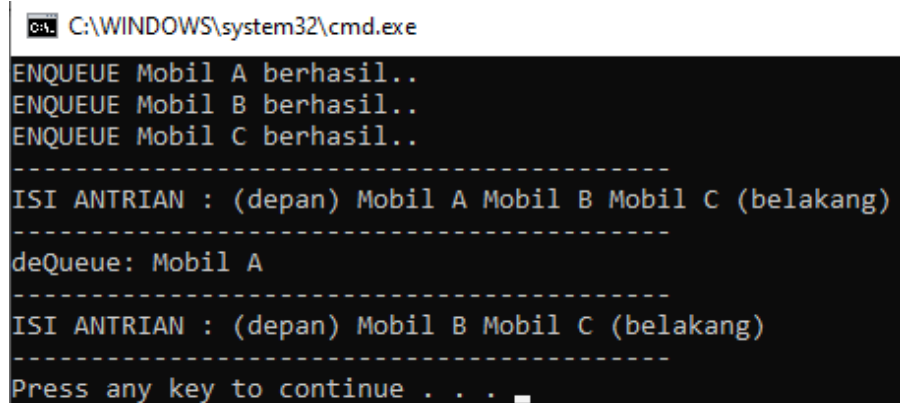
public class P8{
    public static int N = 5;
    public static int belakang = -1;
    public static void enqueue (String antrian[], String data){
        if (belakang == N-1){
            //jika antrian penuh
            System.out.println("maap, antrian penuh, ENQUEUE "
+ data + " tidak dapat dilakukan");
        }else{ //jika antrian tidak penuh
            belakang = belakang + 1;
            antrian[belakang] = data;
            System.out.println("ENQUEUE " + data + " berhasil..");
        }
    }
    public static String dequeue (String antrian[]){
        String hasil;
        if (belakang < 0 ){ //jika antrian kosong

```

```

                                hasil = "ANTRIAN KOSONG, DEQUEUE GAGAL
DILAKUKAN";
                                }else{ //jika antrian tidak kosong
                                    hasil = antrian[0];
                                    //----menggeser data kedua dst, maju selangkah ke
depan
                                    for (int i=0; i<=belakang-1; i++){
                                        antrian[i] = antrian[i+1];
                                    }
                                    belakang = belakang - 1;
                                }
                                return (hasil);
                            }
                        public static void lihatAntrian(String antrian[]){
                            System.out.println("-----");
                            System.out.print("ISI ANTRIAN : (depan)");
                            for (int i=0; i<=belakang; i++){
                                System.out.print(" " + antrian[i]);
                            }
                            System.out.println(" (belakang)");
                            System.out.println("-----");
                        }
                    public static void main (String[] args){
                        String antrian[] = new String[5];
                        enqueue (antrian, "Mobil A");
                        enqueue (antrian, "Mobil B");
                        enqueue (antrian, "Mobil C");
                        lihatAntrian(antrian);
                        System.out.println("deQueue: " + dequeue(antrian));
                        lihatAntrian(antrian);
                    }
                }
    
```

Output :



```

C:\WINDOWS\system32\cmd.exe
ENQUEUE Mobil A berhasil..
ENQUEUE Mobil B berhasil..
ENQUEUE Mobil C berhasil..
-----
ISI ANTRIAN : (depan) Mobil A Mobil B Mobil C (belakang)
-----
deQueue: Mobil A
-----
ISI ANTRIAN : (depan) Mobil B Mobil C (belakang)
-----
Press any key to continue . . .
    
```

Mobil yang terDequeue adalah mobil A dikarenakan mobil A berada di posisi depan sehingga yang terdequeue adalah mobil A. Antrian juga berubah karena mobil A sudah terdequeue sehingga di dalam antrian tinggal mobil B dan mobil C dan yang menjadi mobil belakang adalah C. Program yang dapat merubah itu adalah pada **class String dequeue**.

i. Praktik 9

Sekarang, lakukan lagi proses deQueue dari antrian sebanyak 3 kali berturut-turut :

```
System.out.println("deQueue: " + deQueue(antrian));
System.out.println("deQueue: " + deQueue(antrian));
System.out.println("deQueue: " + deQueue(antrian));
lihatAntrian(antrian);
```

Catatlah “Mobil” apakah saja yang ter-deQueue ? Sekarang amatilah kondisi antrian saat ini. Adakah perubahan yang terjadi pada kondisi antrian saat ini dibanding dengan pada percobaan sebelumnya? Apakah setiap data “Mobil B,C,D,...dst” masih berada di posisinya yang sama? Mengapa hal itu bisa terjadi? Bagaimanakah juga dengan kondisi ekor saat ini? Jelaskan pada laporan anda.

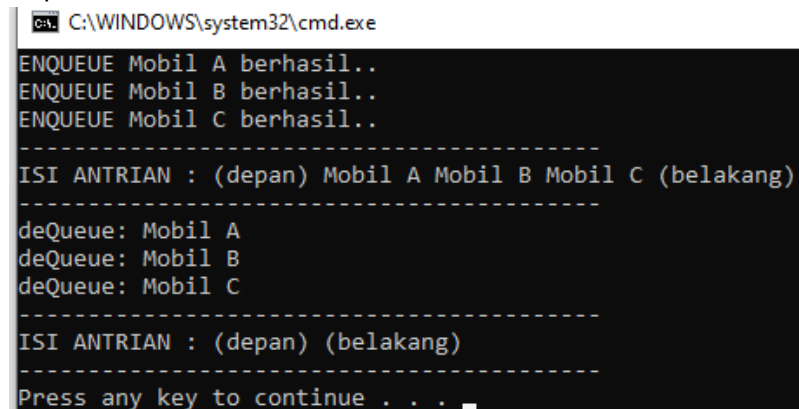
```
public class P9{
    public static int N = 5;
    public static int belakang = -1;
    public static void enqueue (String antrian[], String data){
        if (belakang == N-1){
            //jika antrian penuh
            System.out.println("maap, antrian penuh, ENQUEUE "
+ data + " tidak dapat dilakukan");
        }else{ //jika antrian tidak penuh
            belakang = belakang + 1;
            antrian[belakang] = data;
            System.out.println("ENQUEUE " + data + " berhasil..");
        }
    }
    public static String dequeue (String antrian[]){
        String hasil;
        if (belakang < 0 ){ //jika antrian kosong
            hasil = "ANTRIAN KOSONG, DEQUEUE GAGAL
DILAKUKAN";
        }else{ //jika antrian tidak kosong
            hasil = antrian[0];
            //----menggeser data kedua dst, maju selangkah ke
            depan
            for (int i=0; i<=belakang-1; i++){
                antrian[i] = antrian[i+1];
            }
            belakang = belakang - 1;
        }
        return (hasil);
    }
    public static void lihatAntrian(String antrian[]){
        System.out.println("-----");
        System.out.print("ISI ANTRIAN : (depan)");
        for (int i=0; i<=belakang; i++){
            System.out.print(" " + antrian[i]);
        }
        System.out.println(" (belakang)");
        System.out.println("-----");
    }
}
```

```

    }
    public static void main (String[] args){
        String antrian[] = new String[5];
        enqueue (antrian, "Mobil A");
        enqueue (antrian, "Mobil B");
        enqueue (antrian, "Mobil C");
        lihatAntrian(antrian);
        System.out.println("deQueue: " + dequeue(antrian));
        System.out.println("deQueue: " + dequeue(antrian));
        System.out.println("deQueue: " + dequeue(antrian));
        lihatAntrian(antrian);
    }
}

```

Output :



```

C:\WINDOWS\system32\cmd.exe
ENQUEUE Mobil A berhasil..
ENQUEUE Mobil B berhasil..
ENQUEUE Mobil C berhasil..
-----
ISI ANTRIAN : (depan) Mobil A Mobil B Mobil C (belakang)
-----
deQueue: Mobil A
deQueue: Mobil B
deQueue: Mobil C
-----
ISI ANTRIAN : (depan) (belakang)
-----
Press any key to continue . . .

```

Semua mobil sudah terdequeue sehingga isi dari antrian sudah kosong.

j. Praktik 10

Sekarang, lakukan lagi proses dequeue dari antrian sebanyak 2 kali berturut-turut.

```

System.out.println("deQueue: " + dequeue(antrian));
System.out.println("deQueue: " + dequeue(antrian));
lihatAntrian(antrian);

```

Adakah proses dequeue yang tidak dapat dilakukan? Mengapa bisa demikian? Jelaskan pada laporan anda.

```

public class P10{
    public static int N = 5;
    public static int belakang = -1;
    public static void enqueue (String antrian[], String data){
        if (belakang == N-1){
            //jika antrian penuh
            System.out.println("maap, antrian penuh, ENQUEUE "
+ data + " tidak dapat dilakukan");
        }else{ //jika antrian tidak penuh
            belakang = belakang + 1;
            antrian[belakang] = data;
            System.out.println("ENQUEUE " + data + " berhasil..");
        }
    }
    public static String dequeue (String antrian[]){
        String hasil;
    }
}

```



```

        if (belakang < 0){ //jika antrian kosong
            hasil = "ANTRIAN KOSONG, DEQUEUE GAGAL
DILAKUKAN";
        }else{ //jika antrian tidak kosong
            hasil = antrian[0];
            //----menggeser data kedua dst, maju selangkah ke
            depan
            for (int i=0; i<=belakang-1; i++){
                antrian[i] = antrian[i+1];
            }
            belakang = belakang - 1;
        }
        return (hasil);
    }
    public static void lihatAntrian(String antrian[]){
        System.out.println("-----");
        System.out.print("ISI ANTRIAN : (depan)");
        for (int i=0; i<=belakang; i++){
            System.out.print(" " + antrian[i]);
        }
        System.out.println(" (belakang)");
        System.out.println("-----");
    }
    public static void main (String[] args){
        String antrian[] = new String[5];
        enqueue (antrian, "Mobil A");
        enqueue (antrian, "Mobil B");
        enqueue (antrian, "Mobil C");
        lihatAntrian(antrian);
        System.out.println("deQueue: " + dequeue(antrian));
        System.out.println("deQueue: " + dequeue(antrian));
        lihatAntrian(antrian);
    }
}

```

Output :

```

C:\WINDOWS\system32\cmd.exe
ENQUEUE Mobil A berhasil..
ENQUEUE Mobil B berhasil..
ENQUEUE Mobil C berhasil..
-----
ISI ANTRIAN : (depan) Mobil A Mobil B Mobil C (belakang)
-----
deQueue: Mobil A
deQueue: Mobil B
-----
ISI ANTRIAN : (depan) Mobil C (belakang)
-----
Press any key to continue . . .

```

Semua proses deQueue dapat dilakukan dikarenakan yang ingin di deQueue berjumlah 2 sedangkan mobil yang ada di antrian berjumlah 3 sehingga masih ada sisa data.

C. Kesimpulan

Setelah melakukan praktik di atas sesuai dengan tujuan dari modul 6 ini dapat disimpulkan bahwa, mahasiswa dapat mengimplementasikan tumpukan dan antrian untuk berbagai keperluan dengan menggunakan bahasa pemrograman java.