



# **PEMOGRAMAN WEB II**

Program Studi PJJ INFORMATIKA

**Sesi 4 – QUEUE**

**CATUR NUGROHO, S.KOM., M.KOM**

- Queue (antrian) adalah suatu struktur data dengan penyisipan di satu ujung, sedangkan penghapusan di ujung lainnya.
- Queue atau antrian adalah suatu koleksi item berurut dimana penambahan item baru terjadi di satu ujung bernama “ekor” (rear) dan penghapusan terjadi pada ujung lainnya yang dinamakan “kepala” (front).
- Queues menggunakan pengurutan FIFO, (*First In First Out*), elemen yang pertama masuk akan pertama ke luar

Berikut ini adalah beberapa operasi Queue :

- Operasi penghapusan dilakukan pada ujung front/ head.
- Operasi penyisipan dilakukan pada ujung rear/ tail.
- Front/ head menunjuk ke awal antrian (elemen terdepan), sedangkan rear/ tail menunjuk ke akhir antrian (elemen terakhir).

Rear (terakhir)

Front (terdepan)





***queue()*** membuat suatu antrian baru yang kosong. Tidak memerlukan parameter dan mengembalikan suatu antrian kosong.

***enqueue(item)*** menambahkan suatu item baru ke ujung saru antrian. Perlu item dan tidak mengembalikan sesuatu.

***dequeue()*** menghapus item depan dari antrian. Tidak memerlukan parameter dan mengembalikan itemnya. Antrian termodifikasi.

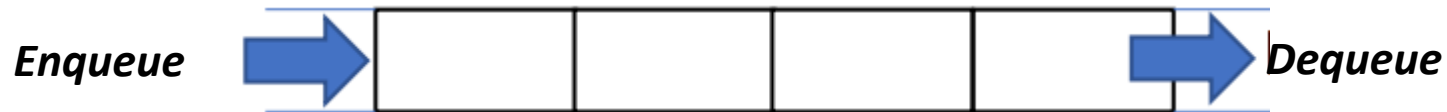
***isEmpty()*** menguji untuk melihat apakah antrian dalam keadaan kosong. Tidak perlu parameter & mengembalian nilai boolean.

***size()*** mengembalikan jumlah item yang ada di dalam antrian. Tidak memerlukan parameter dan mengembalikan suatu integer.



# QUEUE

- Antrean apa pun adalah antrean:
  - Antrean check-out di toko bahan makanan
  - Mobil-mobil di lampu berhenti
  - Jalur perakitan
- Operasi *Enqueue* → tambahkan elemen = Push
- Operasi *Dequeue* → elemen dihapus = Pop





## ***Queue pada computer science***

### ***Operating systems :***

- antrian pekerjaan cetak untuk dikirim ke printer
- antrian program / proses yang akan dijalankan
- antrian paket data jaringan untuk dikirim

### ***Pemrograman:***

- memodelkan garis pelanggan atau klien
- menyimpan antrian komputasi yang akan dilakukan secara berurutan

### ***Pada Contoh dunia nyata:***

- orang-orang di eskalator atau sedang mengantre
- mobil di pom bensin (atau di jalur perakitan)

## Simulasi: Antrian Mencetak dalam Jaringan

Langkah-langkah utama dari simulasi antrian mencetak ke suatu printer dari banyak komputer

adalah membuat suatu antrian (queue) tugas (task) mencetak. Setiap task akan diberikan suatu timestamp saat kedatangannya. Queue dimulai dalam keadaan kosong.

1. Untuk setiap waktu (*currentSecond*): Apakah suatu tugas cetak akan dibuat? Jika Iya, maka tambahkan itu ke queue dengan *currentSecond* sebagai timestampnya.
2. Jika printer tidak sibuk dan jika suatu task sedang menunggu, hapus task berikutnya dari antrian cetak (*print queue*) dan serahkan task tersebut ke printer.
3. Kurangkan timestamp dari *currentSecond* untuk menghitung waktu tunggu (*waiting time*) untuk task tersebut.
4. Tambahkan *waiting time* untuk task tersebut ke suatu list untuk pemrosesan akan datang.



## Simulasi: Antrian Mencetak dalam Jaringan

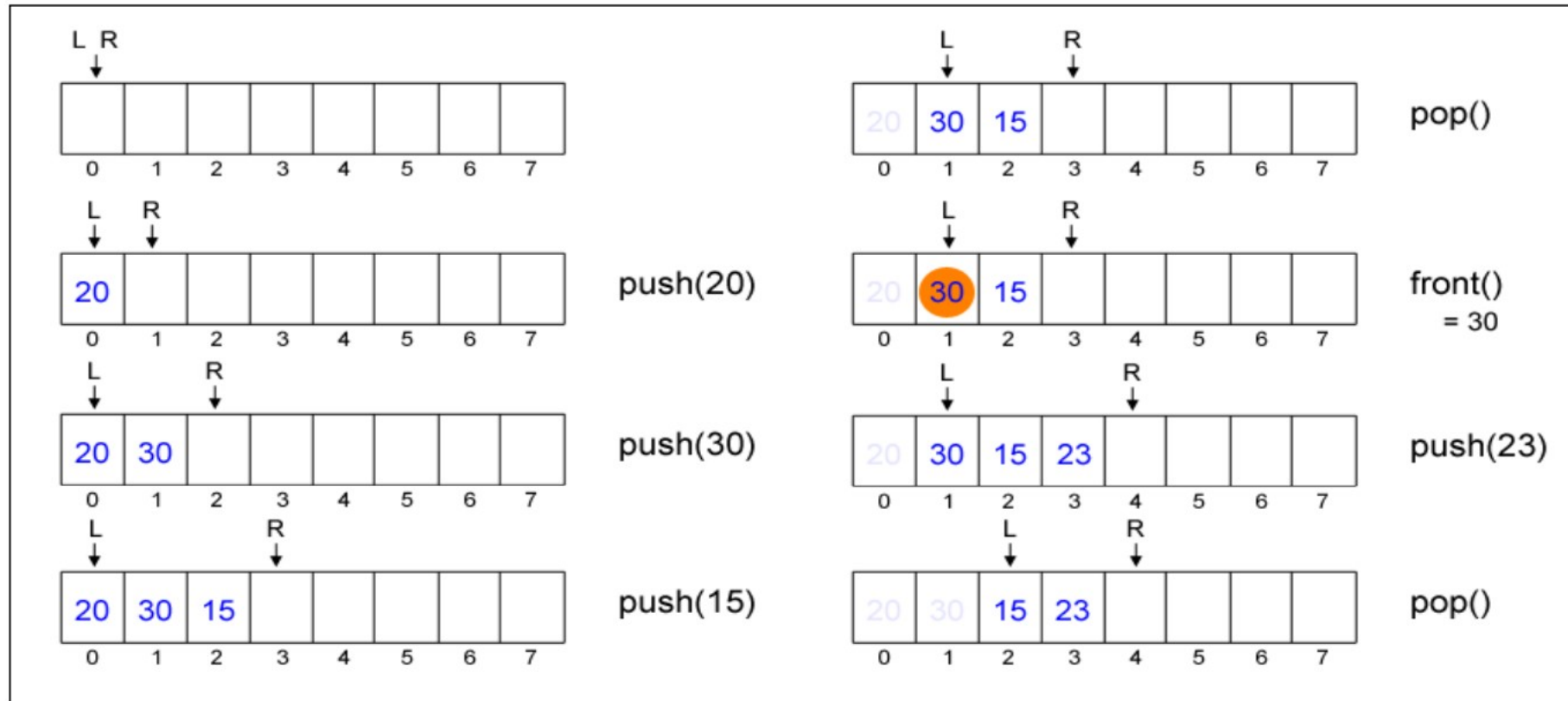
5. Berdasarkan pada jumlah halaman dalam print task, tetapkan berapa waktu yang diperlukannya.
6. Printer sekarang melakukan satu detik pencetakan jika diperlukan. Ia juga mengurangi satu detik dari waktu yang diperlukan untuk tugas tersebut.
7. Jika task tersebut telah selesai (*completed*), dengan kata lain waktu yang diperlukan telah mencapai nol, maka printer tidak sibuk lagi.
8. Setelah simulasi selesai, hitung waktu tunggu rata-rata (*average waiting time*) dari daftar waktu tunggu yang dibangkitkan.





# QUEUE

## Contoh Operasi Queue :





## Array Representation :

Antrian memiliki dua variabel:

Depan dan belakang yang mengarah ke posisi tempat penghapusan dan penyisipan dapat dilakukan masing-masing

➤ Contoh :

➤ Di sini, depan = 0 dan belakang = 5.

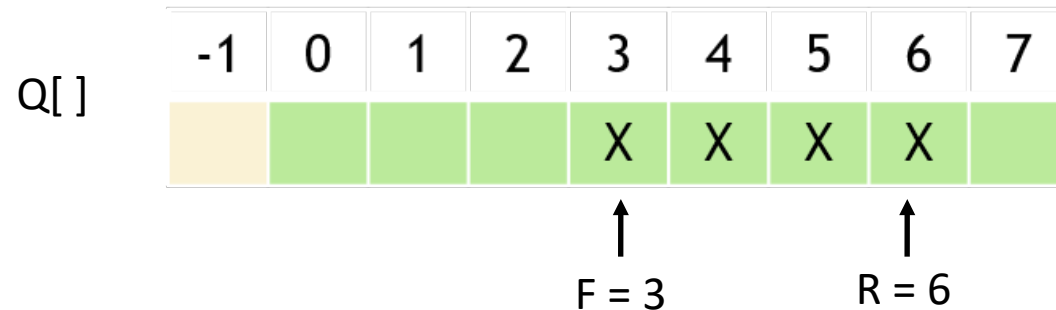


# QUEUE

1. *Linear Queue* (antrian lurus)
2. *Circular Queue* (antrian melingkar)
3. *Double Ended Queue* (antrian dengan ujung ganda)

## I. LINIER QUEUE (antrian lurus)

Jika sebuah array terdapat 8 elemen ( $n=8$ ), maka Queue dapat diilustrasikan sbb:



Ada dua indeks yang digunakan yaitu F & R  
sbb:

F = *front*, menunjukkan posisi terdepan.

R = *rear*, menunjukkan posisi terakhir.



## I. LINIER QUEUE (antrian lurus)

### Proses :

1. *Inisialisasi* (proses awal)  $\rightarrow F = R + 1$

Jika array belum ada isinya (kosong), maka  $F = 0$  dan  $R = -1$  ( $F > R$ )

2. *Insert* (simpan)  $\rightarrow R = R + 1; Q[R] = X$

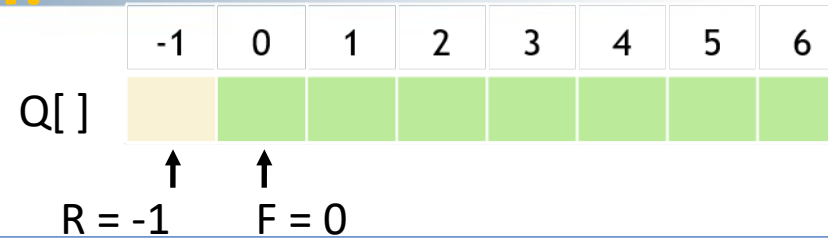
3. *Delete* (hapus)  $\rightarrow X = Q[F]; F = F + 1$

4. *Reset* (Kembali ke keadaan awal/ kosong)  $\rightarrow F = 0; R = -1$



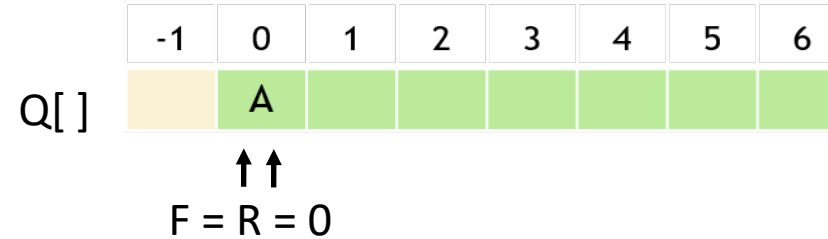
# QUEUE Contoh:

- 1) Kondisi awal.  
Array 7 elemen  
(n=7)



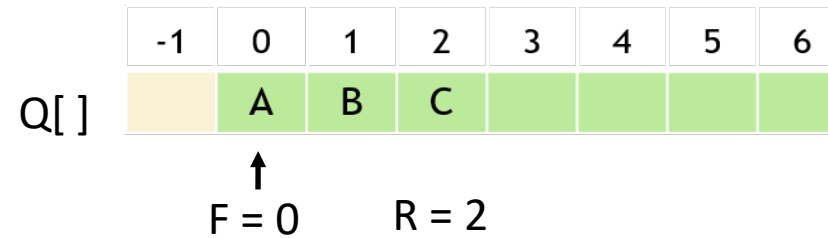
$F = R + 1$  antrian kosong  
 $R < n - 1$  antrian bisa diisi

- 2) Insert A.



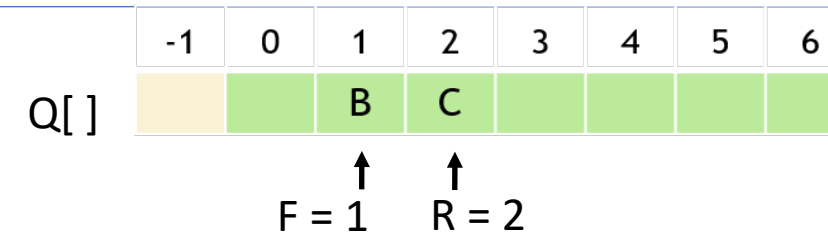
$F < R + 1$  antrian ada isinya  
 $R < n - 1$  antrian bisa diisi  
 $F = \text{Risi antrian ada satu}$   
 $F = 0$  belum ada yg keluar

- 3) Insert B dan C.



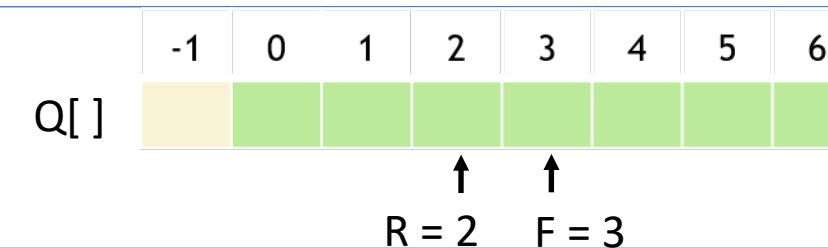
$F < R + 1$  antrian ada isinya  
 $R < n - 1$  antrian bisa diisi  
 $F = 0$  belum ada yg keluar

- 4) Delete A.



$F < R + 1$  antrian ada isinya  
 $R < n - 1$  antrian bisa diisi  
 $F = 1$  antrian sudah keluar 1

- 5) Delete B dan C.



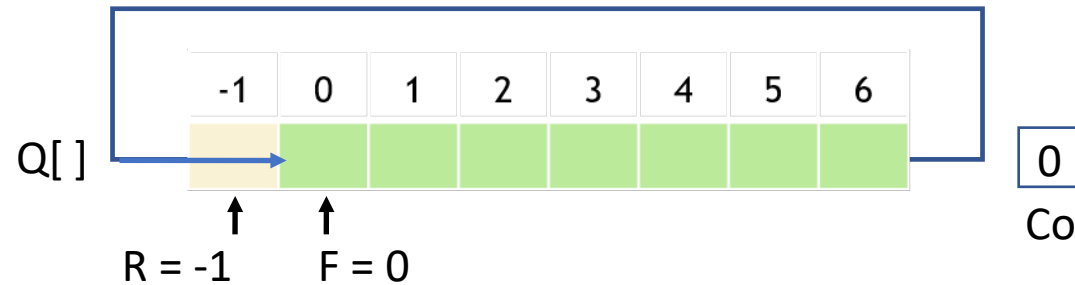
$F = R + 1$  antrian kosong  
 $R < n - 1$  antrian bisa diisi  
 $F = 3$  antrian sudah keluar 3



## 2. CIRCULAR QUEUE (antrian melingkar)

Memanfaatkan tempat yang telah kosong dalam antrian. Jika antrian sudah sampai pada posisi terakhir,  $Q[n-1]$ , maka antrian dapat dilanjutkan ke  $Q[0]$  dan seterusnya.

Kondisi awal.  
Array 7 elemen  
( $n=7$ )

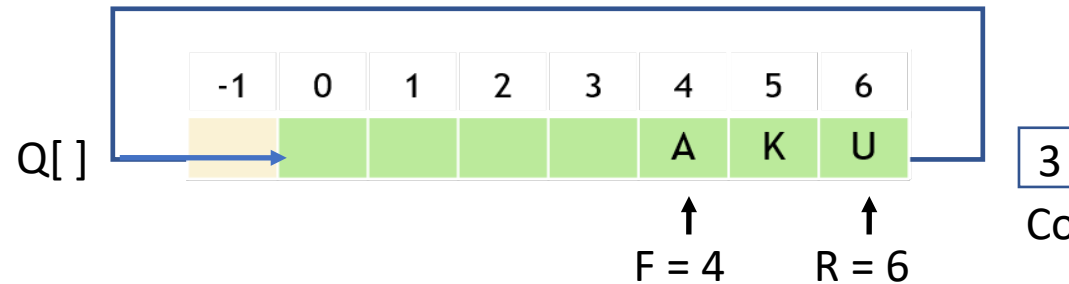


$F = R+1$  antrian kosong  
Counter = 0

Counter = jumlah elemen yang ada.

Contoh:

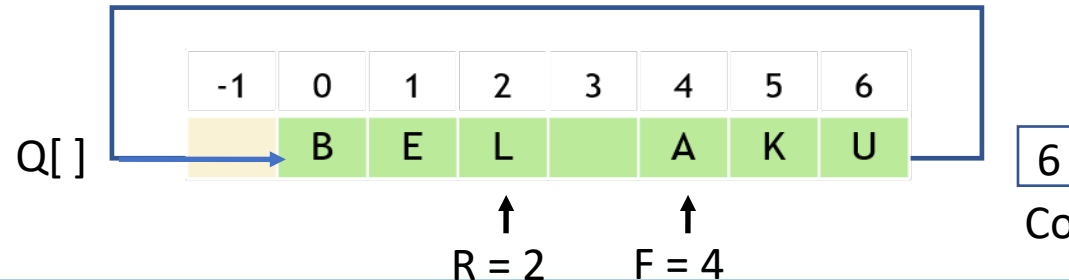
$n=7$



Counter > 0 antrian ada isinya  
Counter < n antrian bisa diisi

3  
Counter

Insert B, E, L.



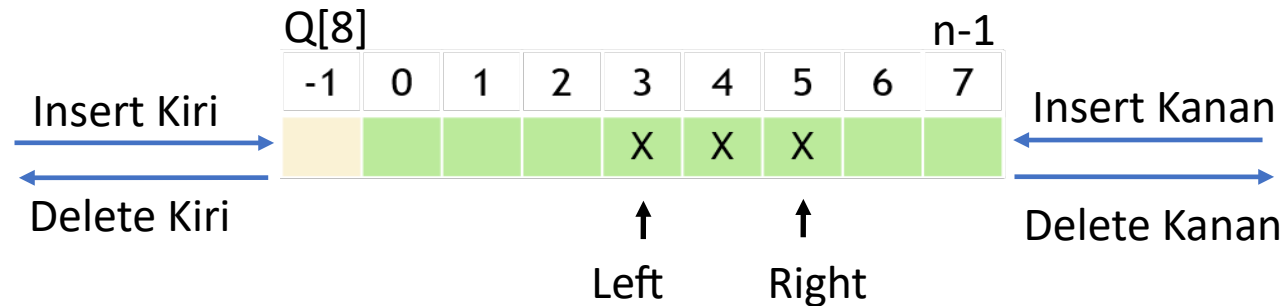
Counter > 0 antrian ada isinya  
Counter < n antrian bisa diisi

6  
Counter



### 3. DOUBLE ENDED QUEUE (DEQUE)

(antrian dengan ujung ganda)



- Terdapat dua index, yaitu Left dan Right.
- Proses: *Awal, Insert, Delete*.
- Elemen data dapat masuk dan keluar dari sisi kiri maupun sisi kanan, tergantung kesempatan yang ada. Tetapi tidak dapat dilakukan di tengah-tengah.





# QUEUE

Pada ilustrasi gambar antrian di atas:

Index L menunjuk  $Q[3]$  dan Index R menunjuk  $Q[5]$ .

Proses:

Insert Left → data masuk di  $Q[2]$ ;  $L = L - 1$  dan  $Q[L] = X$

Insert Right → data masuk di  $Q[6]$ ;  $R = R + 1$  dan  $Q[R] = X$

Delete Left → hapus isi data  $Q[3]$ ;  $X = Q[L]$  lalu  $L = L + 1$ , menunjuk  $Q[4]$

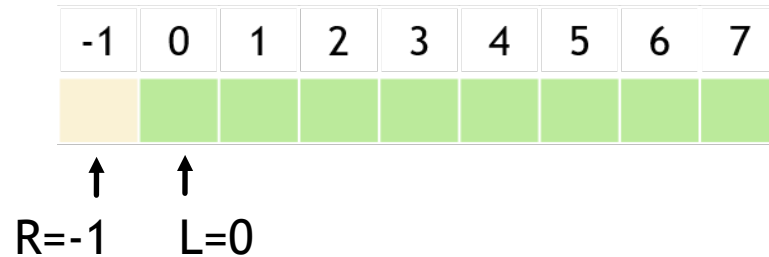
Delete Right → hapus isi data  $Q[5]$ ;  $X = Q[R]$  lalu  $R = R - 1$ , menunjuk  $Q[4]$



### 3. DOUBLE ENDED QUEUE (DEQUE)

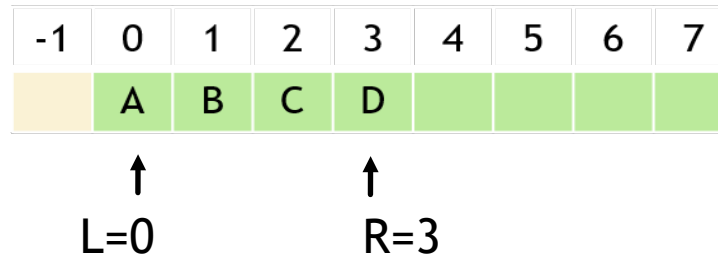
(antrian dengan ujung ganda)

1) Kondisi awal.

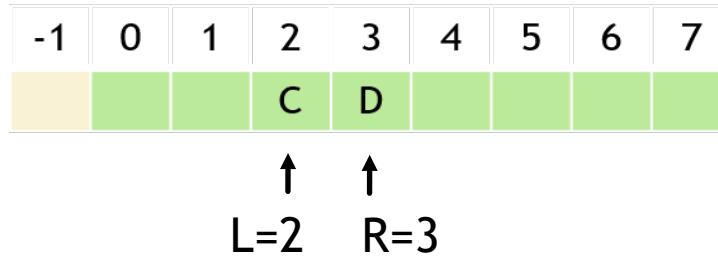


Khusus dalam kondisi antrian kosong, proses yang bisa dilakukan hanya Insert Right.

2) Insert Right A, B, C, D.



3) Delete Left A, B.



4) Delete Right D.

