



Queue

(TIB11 – Struktur Data)

Pertemuan 17, 18

Sub-CPMK

- Mahasiswa mampu menggunakan linked list dan array untuk membuat queue beserta operasi-operasinya (C3, A3)

Materi

- Pengertian Queue
- Array Base Queue
- Linked-List Base Queue

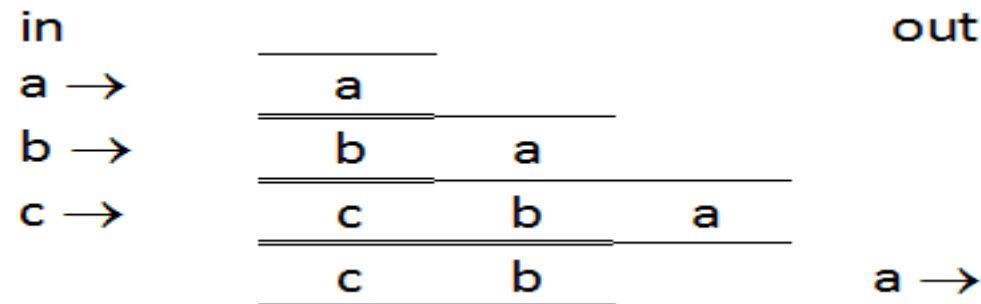


1.

Pengertian Queue

Queue

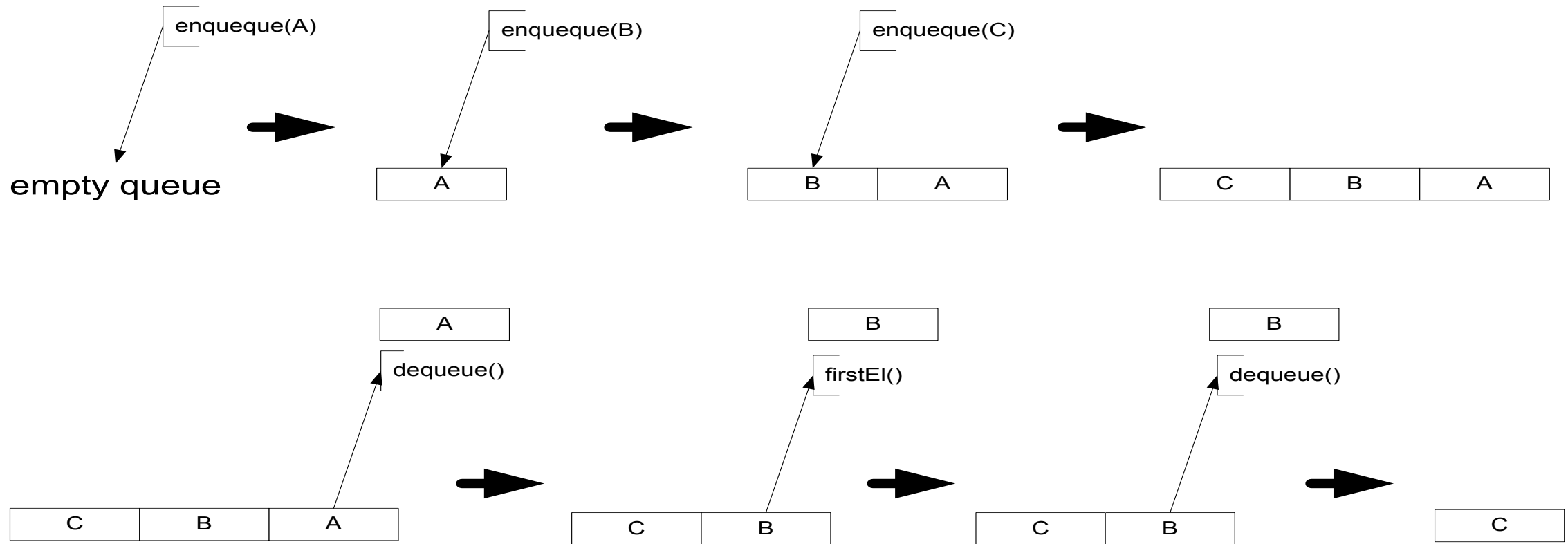
- Penambahan elemen dengan cara penambahan elements pada akhir dan mengeluarkan element dari depan
- First In First Out



Queue Operation

- `clear()` → menghapus / membersihkan queue
- `isEmpty()` → memeriksa apakah queue kosong
- `enqueue(el)` → memasukkan element el pada akhir queue
- `dequeue` → Mengambil element pertama dari queue
- `firstEl()` → Membaca first element dari queue tanpa menghapusnya

Queue Operation (cont.)





2.

Array Base Queue

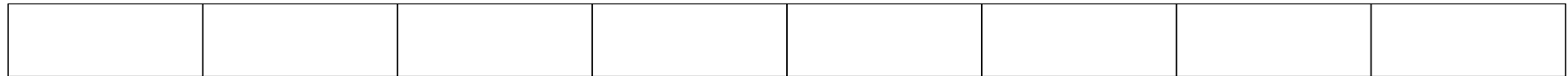
Implementasi Queue dengan Array

Variable yang dibutuhkan

- Array sebagai queue pool
- Integer variable First untuk menginformasikan offset number dari array yang menjadi Queue pertama
- Integer variable Last untuk menginformasikan offset number dari array yang menjadi Queue terakhir
- array harus circular untuk mempermudah penerapan

Queue Implementation with Array

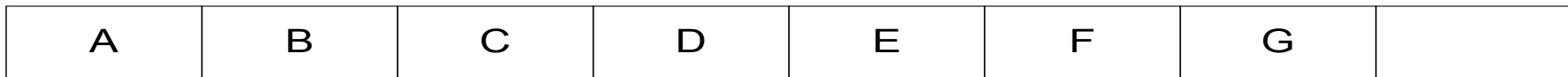
First Last



First



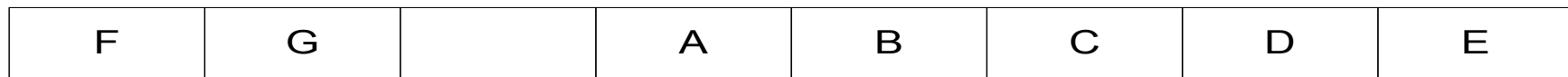
Last



Last



First



Kondisi First dan Last Variable pada circular array

- Pada kondisi awal, kedua First dan Last variable dapat di set -1 untuk mengindikasikan queue is empty dan data size harus di set dengan 0
- Setelah queue terisi, normally
If empty() then First=0 //set to the 1st array
Last = First + Data Size - 1
- Last variable dapat berisi nilai yang lebih rendah daripada First jika First value lebih dari 0 dan First + Data Size – 1 lebih besar dari Array Size.
- Kedua kondisi di atas dapat diterapkan dengan notasi
Last = ((First + Data Size - 1) mod Array Size)

Queue operation with array

- `clear()` →
 - Isi First dan Last dengan -1 value dan data size dengan 0
- `isEmpty()` →
 - Periksa First or Last value, jika berisi -1 maka berarti queue kosong,
 - Atau periksa data size, jika berisi 0 maka queue kosong

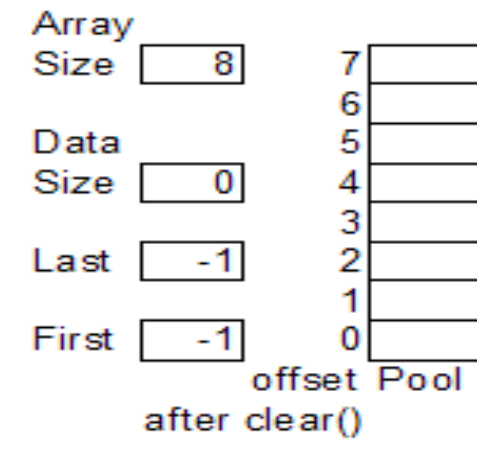
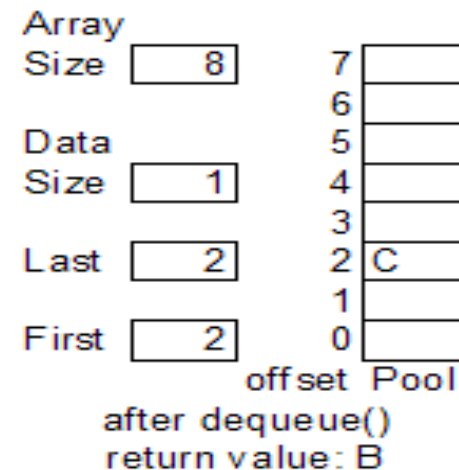
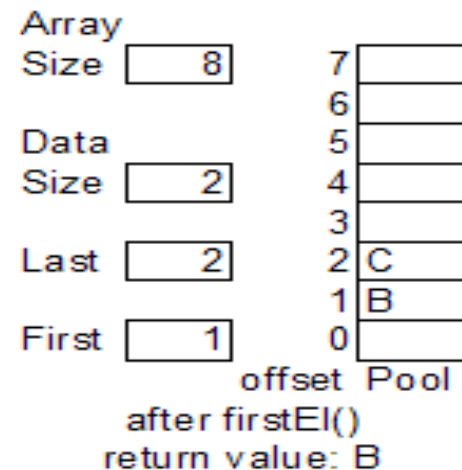
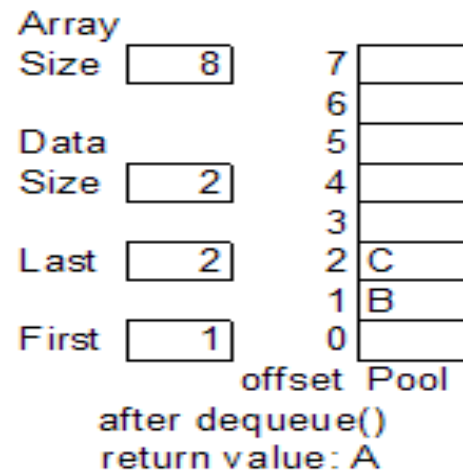
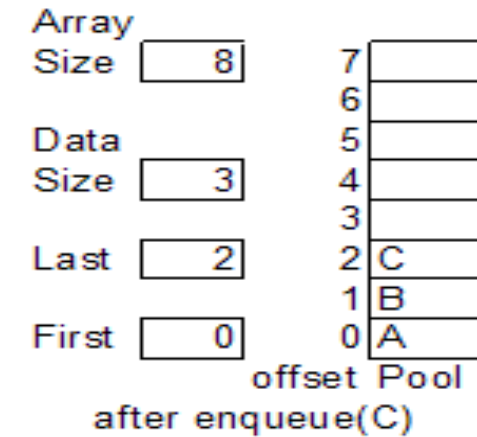
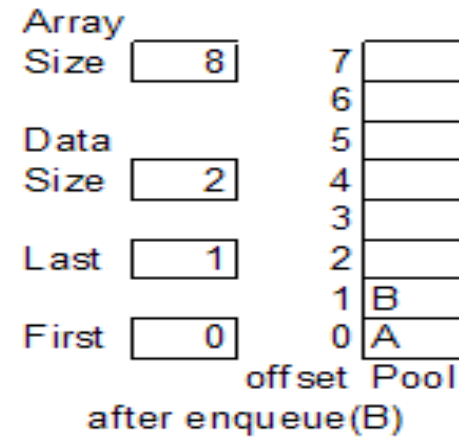
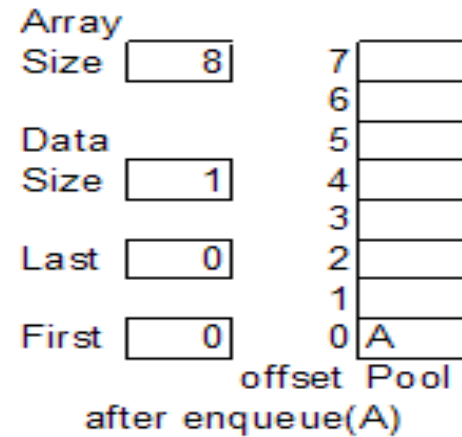
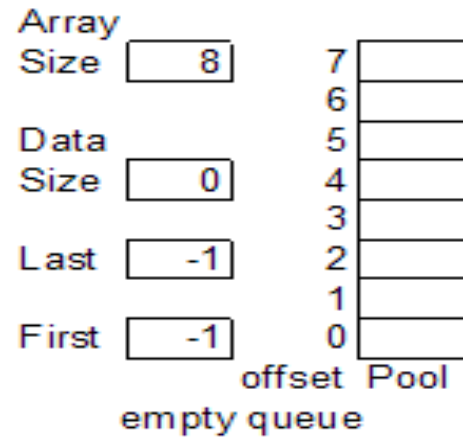
Queue operation with array (cont.)

- enqueue(el) →
 - Set First value dengan 0 jika kosong.
if isEmpty() then First = 0;
 - Increment Data Size value
Inc(Data Size);
 - Set Last value dengan
 $\text{Last} = (\text{First} + \text{Data Size} - 1) \bmod \text{Array Size}$
 - Kemudian masukkan element el ke sel array pada offset Last
 $\text{Pool}[\text{Last}] = \text{el}$

Queue operation with array (cont.)

- dequeue →
 - Ambil value dari Pool[First]
 - Hapus data pada Pool[First]
 - Decrement DataSize
 - Set next value dari variabel First
$$\text{First} = (\text{First} + 1) \bmod \text{ArraySize}$$
- firstEl() →
 - Ambil data yang terdapat Pool[First]

Queue operation with array (cont.)





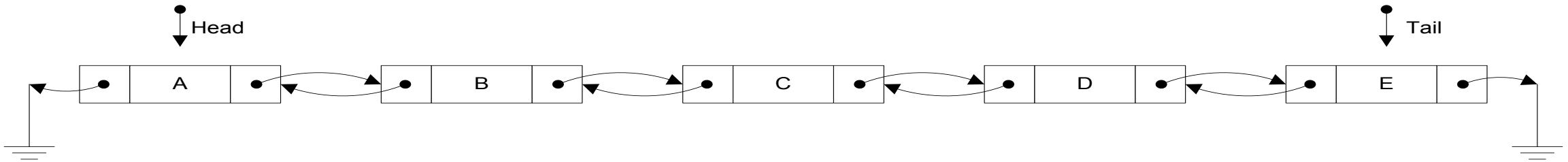
3.

Linked-List Base Queue

Penerapan Queue dengan Linked List

- Hanya diperlukan Double Linked List dan cara yang mudah untuk mendapatkan queue pertama dan terakhir
- (jika menggunakan single linked list, akan sedikit lebih repot pada saat melakukan dequeue, pada first step harus menyimpan current head ke temporary variable, untuk membebaskan current head dengan mudah)
- First queue dapat diterapkan dengan head
- Last queue dapat diterapkan dengan tail
(atau kebalikannya: head sebagai Last queue dan tail sebagai First queue, tapi harus punya pointer untuk menunjuk ke previous tail – dapat diterapkan dengan double linked list or single list dengan informasi Previous Tail)

Queue Implementation with Linked List (cont.)

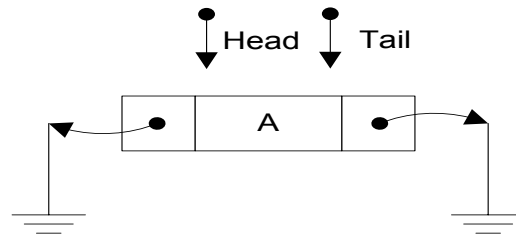


Queue Operation dengan linked list

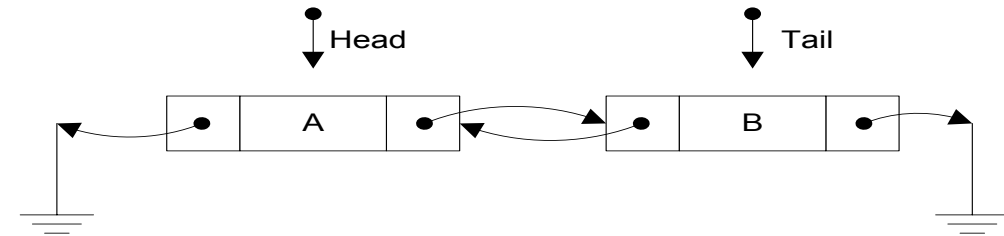
- `clear()` →
 - Hancurkan setiap node
 - set Head dengan NULL
- `isEmpty()` →
 - Periksa jika `Head == NULL`, maka queue is empty
- `enqueue(el)` →
 - Buat node baru pada tail,
 - Isi element `el` pada node baru.
- `Dequeue` →
 - Ambil element dari node yang ditunjuk oleh Head,
 - set next Node sebagai Head
 - destroy node yang semula Head
- `firstEl()` →
 - Ambil element dari node Head.

Queue Operation with linked list (cont.)

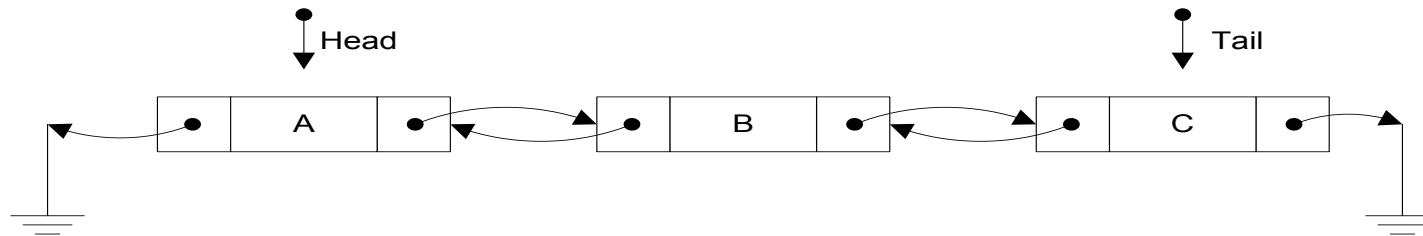
Empty queue



After enqueue(A)

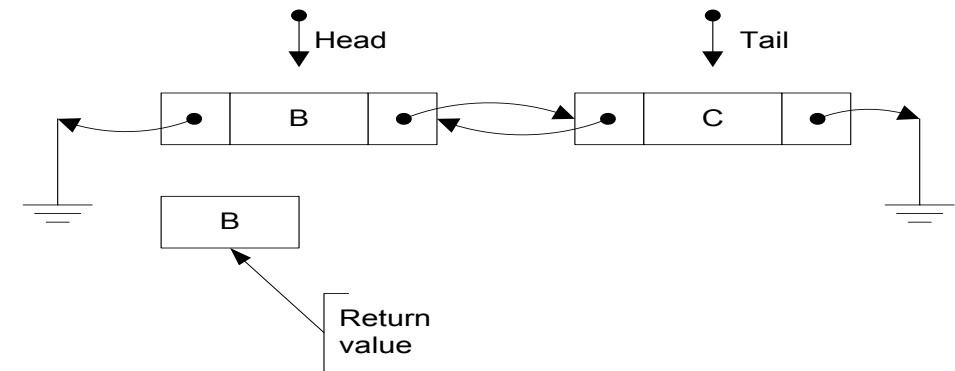
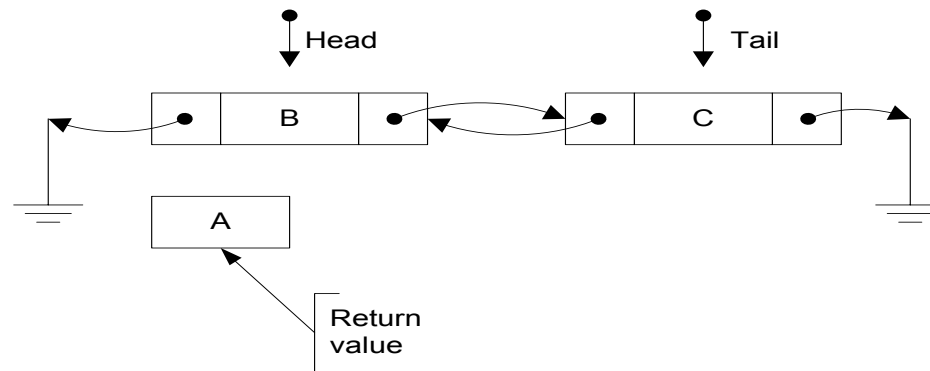
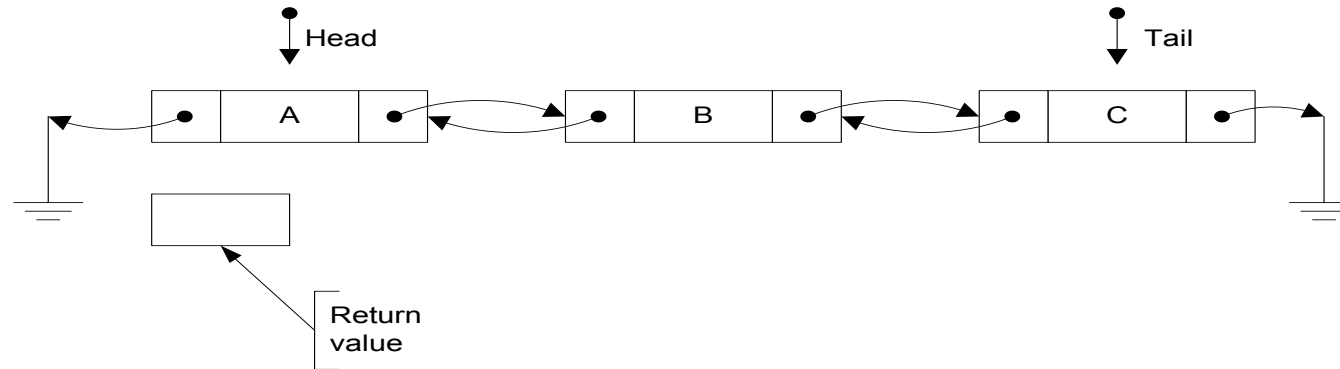


After enqueue(B)



After enqueue(C)

Queue Operation with linked list (cont.)



`dequeue()`

`firstEl()`

Tambahan

- Untuk penggunaan Single List, ketika melakukan dequeue,
 - anda harus menyimpan dahulu alamat head saat ini pada variabel temp.
 - Setelah data diambil dari queue, arahkan head ke node berikutnya.
 - Dan free kan node yang sebelumnya merupakan head yang sebelumnya yang sekarang ditunjuk oleh variable temp di atas
- Untuk penggunaan double linked list, anda cukup mengarahkan head ke node berikutnya, kemudian free kan node yang sebelumnya merupakan head yang sekarang ditunjuk oleh Head->Prev

Ringkasan

- Queue adalah Struktur Data Linear dimana penambahan elemen dilakukan pada akhir dan mengeluarkan element dari depan (First In First Out). berisi operasi : `clear()`, `isEmpty()`, `enqueue(el)`, `dequeue()` dan `firstEl()`
- Queue dapat diterapkan dengan dua cara
 - Linked-List Base Queue--> implementasi queue menggunakan linked-list
 - Array Base Queue --> implementasi queue menggunakan array



Terimakasih

TUHAN Memberkati Anda

Teady Matius Surya Mulyana (tmulyana@bundamulia.ac.id)