



TIB29 – Struktur Data dan Algoritma

PERINGATAN HAK CIPTA

Segala materi ini merupakan milik Universitas Bunda Mulia yang dilindungi oleh hak cipta.

Materi ini hanya untuk dipergunakan oleh mahasiswa Universitas Bunda Mulia dalam rangkaian proses perkuliahan.

Dilarang keras untuk mendistribusikannya dalam bentuk apapun.

Pelanggaran terhadap hak cipta ini dapat dikenakan sanksi hukum sesuai dengan perundang-undangan yang berlaku.

© Universitas Bunda Mulia

PERINGATAN HAK CIPTA

Segala materi ini merupakan milik Universitas Bunda Mulia yang dilindungi oleh hak cipta.

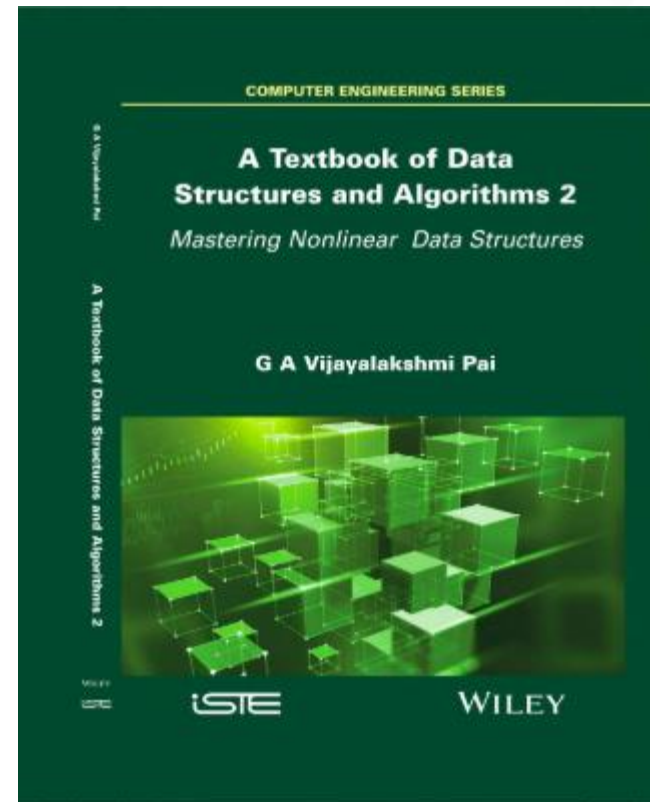
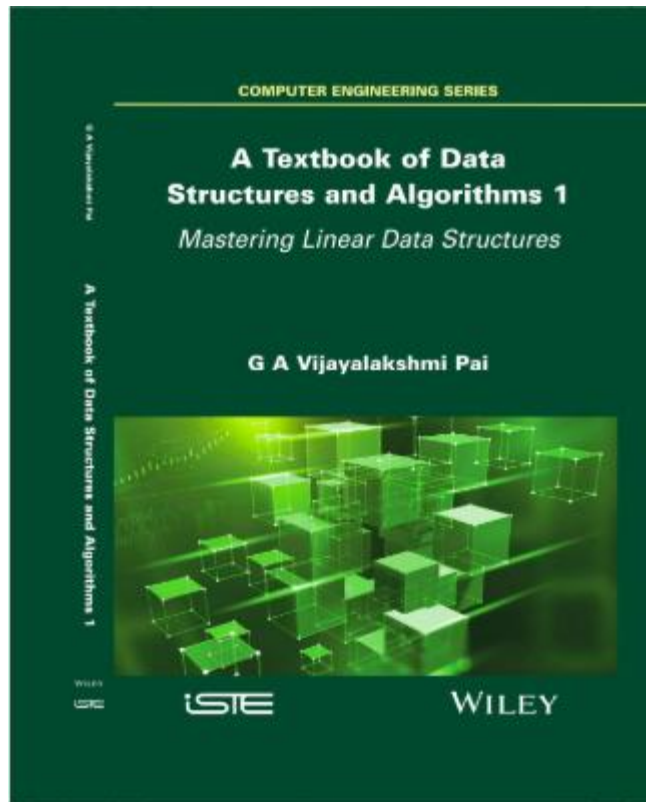
Dilarang keras untuk mengunduh dan atau merekam dan atau mendistribusikannya dalam bentuk apapun.

Materi ini hanya untuk dipergunakan oleh mahasiswa Universitas Bunda Mulia dalam rangkaian proses perkuliahan.

Pelanggaran terhadap hak cipta ini dapat dikenakan sanksi hukum sesuai dengan perundang-undangan yang berlaku

© 2024 Universitas Bunda Mulia

Diadopsi Dari Sumber:





Queue Berbasis Array

Sub-CPMK

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

Materi:

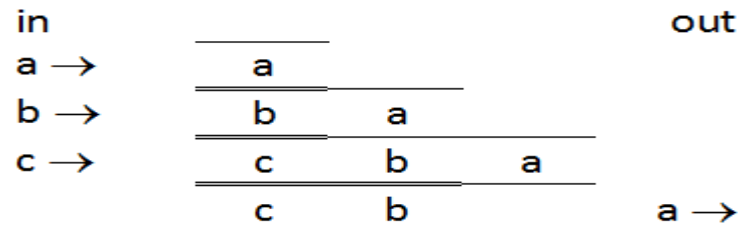
- 1. Pengertian Queue*
- 2. Array Base Queue*
- 3. Operasi-operasi Queue*



1. Pengertian *Queue*

1.1 Queue

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

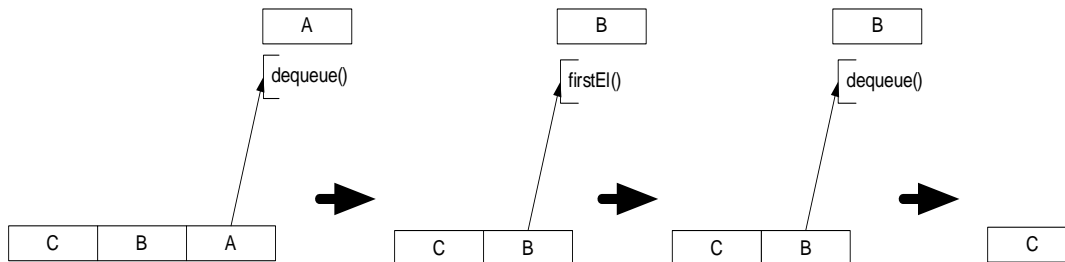


1.2 Queue Operation

- `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

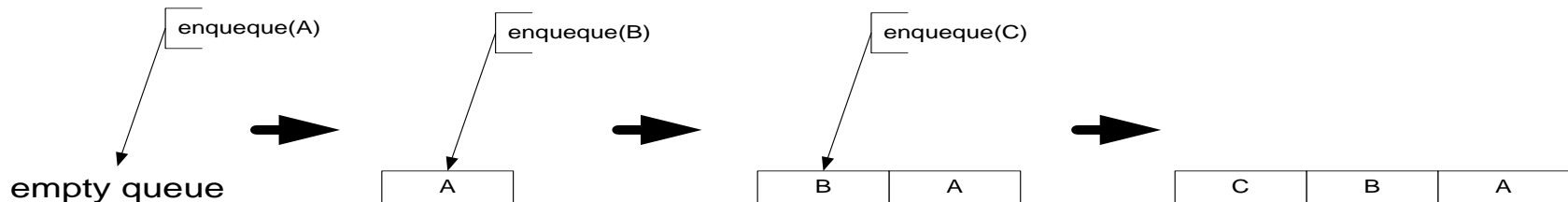
1.2 Queue Operation (Lanj.)

- `clear()` → menghapus / membersihkan *queue*
- *Mulai dari last element, lakukan `dequeue()` sampai seluruh elemen habis*



1.2 Queue Operation (Lanj.)

- enqueue(el) → memasukkan *element* el pada akhir *queue*
- *Enqueue dilakukan pada first queue*





2. Array Base Queue

2.1 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

2.2 Kondisi *First* dan *Last Variable* Pada *Circular Array*

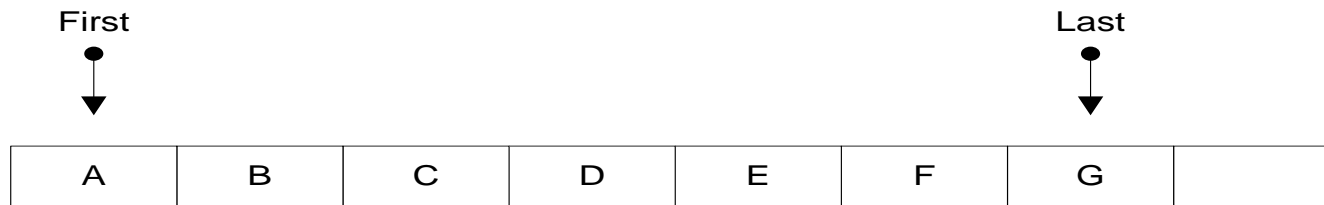
- Pada kondisi awal, kedua *First* dan *Last variable* dapat di set -1 untuk mengindikasikan *queue is empty* dan data size harus di set dengan 0

First Last



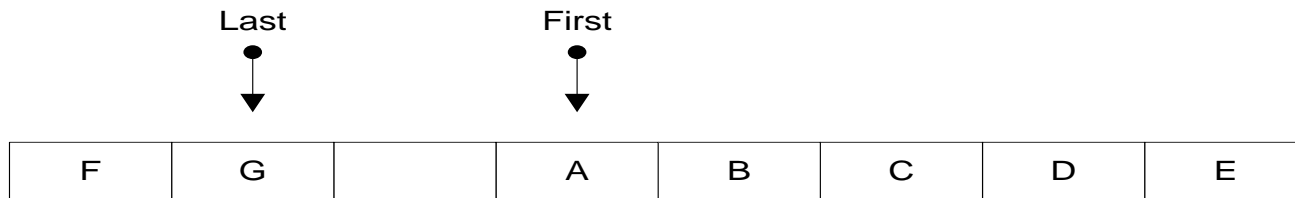
2.2 Kondisi *First* dan *Last Variable* Pada *Circular Array* (Lanj.)

- Setelah *queue* terisi, *normally*
If empty() then First=0 //set to the 1st array
Last = First + Data Size - 1



2.2 Kondisi *First* dan *Last Variable* Pada *Circular Array* (Lanj.)

- *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
 $Last = ((First + Data\ Size - 1) \bmod Array\ Size)$





3. Operasi-operasi Queue

2.3. *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

2.3. Queue Operation With Array (Lanj.)

- 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*
Pool[Last] = el

2.3. Queue Operation With Array (Lanj.)

- 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]

2.4 Penggunaan Python pada Array Base Queue

- Pada pemrograman Python dapat memanfaatkan list sebagai
- Penggunaan list sebagai Array, mengakibatkan Array pada Python berupa Array dinamis
- Karena menggunakan list, maka array base queue menjadi lebih mudah, karena memanfaatkan perintah-perintah list. Tidak perlu menyediakan size dari list, seperti pada Bahasa pemrograman generasi 3, karena dapat mempergunakan perintah `len(namaArray)` untuk mendapatkan Panjang / ukuran dari list yang dimanfaatkan sebagai array

2.4 Penggunaan Python pada Array Base Queue (Lanj.)

Contoh

```
1 def firstEl(pool):  
2     return(pool[0])  
3  
4 #perhatikan, berbeda  
5 #dengan pop pada stack  
6 def deQueue(pool):  
7     return pool.pop(0)  
8  
9 def enqueue(elemen, pool):  
10    pool.append(elemen)  
11  
12 def clearQueue(pool):  
13    pool.clear()  
14  
15 def isEmpty(pool):  
16    return len(pool)==0
```

```
18 #main  
19 dat = [10, 11, 12, 13,  
20        14, 15, 16, 17,  
21        18, 19]  
22 print(dat)  
23 print(firstEl(dat))  
24 print(deQueue(dat))  
25 print(dat)  
26 enqueue(20, dat)  
27 print(dat)  
28 print(isEmpty(dat))  
29 clearQueue(dat)  
30 print(isEmpty(dat))
```

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*

PERINGATAN HAK CIPTA

Segala materi ini merupakan milik Universitas Bunda Mulia yang dilindungi oleh hak cipta.

Dilarang keras untuk mengunduh dan atau merekam dan atau mendistribusikannya dalam bentuk apapun.

Materi ini hanya untuk dipergunakan oleh mahasiswa Universitas Bunda Mulia dalam rangkaian proses perkuliahan.

Pelanggaran terhadap hak cipta ini dapat dikenakan sanksi hukum sesuai dengan perundang-undangan yang berlaku

© 2024 Universitas Bunda Mulia

PERINGATAN HAK CIPTA

Segala materi ini merupakan milik Universitas Bunda Mulia yang dilindungi oleh hak cipta.

Materi ini hanya untuk dipergunakan oleh mahasiswa Universitas Bunda Mulia dalam rangkaian proses perkuliahan.

Dilarang keras untuk mendistribusikannya dalam bentuk apapun.

Pelanggaran terhadap hak cipta ini dapat dikenakan sanksi hukum sesuai dengan perundang-undangan yang berlaku.

© Universitas Bunda Mulia



Terima kasih

TUHAN Memberkati Anda

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