

# Praktikum Struktur Data

## Modul 4 - Queue dan Deque

Senin, 25 April 2022

Tujuan dari modul ini agar mahasiswa memahami konsep struktur Queue dan implementasinya untuk Proses Penjadwalan, serta struktur data Deque. Kerjakan tugas-tugas yang terdapat dalam modul ini, dengan ketentuan sebagai berikut :

1. Semua jawaban modul dikerjakan dalam format \*.ipynb (baik konsep maupun implementasi), jangan lupa diberikan **heading** di setiap cell jawaban, misalkan untuk jawaban konsep Nomor 1, diberikan heading **Konsep\_1**. Silahkan dilihat contoh *lecture notes* yang saya telah bagikan melalui website.
2. Penamaan file ipynb adalah : NPM\_ModulX\_TopikModul.ipynb, misalkan, 200411100077\_Modul4\_QueueDeque.ipynb
3. Print menjadi file pdf, dokumen ipynb tersebut dengan nama yang sama, hanya saja berekstensi pdf, misalkan, 200411100077\_Modul3\_StackInfix.pdf
4. Submit **link collaboratory** yang berisi file ipynb tersebut, dan submit **file pdf**
5. Pilih salah satu nomor untuk dibuat video pembelajaran dan upload ke channel youtube masing-masing, dengan hashtag #PraktikumStrukturDataModul4. Format video bebas, wajib diberikan voice over. Akan lebih baik jika mode camera adalah ON.
6. Kejujuran selalu jadi yang utama, kerjakan sendiri, tidak diperkenankan plagiarisme

## 1 Konsep

Tulis ringkasan atau penjelasan hal-hal berikut, dengan kata-kata kalian sendiri mengenai :

1. Queue, pengertian, dan operasi-operasi yang terdapat pada queue
2. Deque, pengertian, dan operasi-operasi yang terdapat pada Deque

HashtagVideo : Modul4\_QueueDeque dan PraktikumStrukturData\_KelasX

## 2 Implementasi

### 2.1 *Scheduling / Penjadwalan*

Terdapat proses penjadwalan CPU (*resource sharing*). Pada penjadwalan CPU terdapat beberapa proses sebagai berikut :

- Semua **task** masuk kedalam antrian, setiap **task** ini terdapat atribut waktu yang dibutuhkan masing-masing task untuk menggunakan CPU. Misalkan **task A** membutuhkan waktu untuk diproses di CPU selama 7 detik
- **Task** yang berada di posisi paling depan dari antrian, mendapat kesempatan pertama untuk diproses ke dalam CPU

- CPU memiliki atribut waktu, yaitu semua **task** hanya boleh menggunakan CPU selama waktu tertentu. Misalkan waktu CPU adalah 3 detik, maka semua **task** hanya boleh mengakses/menggunakan waktu CPU 3 detik saja
  - Jika suatu **task** membutuhkan waktu lebih dari waktu CPU, maka task tersebut akan diproses selama waktu CPU saja, kemudian task akan dikeluarkan dan dimasukkan kembali (posisi **rear**) ke dalam antrian, agar task tersebut dapat diproses selanjutnya
  - Jika **task** membutuhkan waktu tidak melebihi waktu proses CPU, maka **task** diproses sesuai waktu proses task, kemudian dikeluarkan dari antrian, yang berarti **task** tersebut telah selesai diproses oleh CPU.

Buatlah ilustrasi penjadwalan CPU tersebut dengan menggunakan modul **queue** yang telah dibuat sebelumnya, dengan ketentuan sebagai berikut :

1. Input berupa jumlah \*task\* atau proses yang akan diproses pada CPU
2. Nama proses, beserta atribut waktu yang dibutuhkan masing-masing proses untuk menggunakan CPU
3. Waktu proses CPU

Contoh ilustrasi penjadwalan CPU dapat dilihat pada Gambar 1 HashtagVideo : Modul4.SchedulingQueue dan PraktikumStrukturData.KelasX

Selamat Mengerjakan, Selalu Latihan, Jujur  
 harus dimulai kapanpun, Bertanya jika kurang  
 mengerti, #StayAtHome,  
 #LearningFromHome

---

*Struktur Data - 2022*  
*Indah Agustien Siradjuddin*

---

Jumlah Proses yang akan dijadwal di CPU = 3  
Nama Proses ke-0 : A  
Waktu proses : 5  
Nama Proses ke-1 : B  
Waktu proses : 9  
Nama Proses ke-2 : C  
Waktu proses :

(a)

---

Antrian Proses :  
[['C', 2], ['B', 9], ['A', 5]]

(b)

---

waktu proses CPU = 3  
Antrian Proses beserta Waktunya = [['C', 2], ['B', 9], ['A', 5]]  
Iterasi ke- 1 :  
    Proses A sedang diproses, dan sisa waktu proses A = 2  
    Data proses yang tersisa : [['A', 2], ['C', 2], ['B', 9]]  
Iterasi ke- 2 :  
    Proses B sedang diproses, dan sisa waktu proses B = 6  
    Data proses yang tersisa : [['B', 6], ['A', 2], ['C', 2]]  
Iterasi ke- 3 :  
    Proses C telah selesai diproses  
    Data proses yang tersisa : [['B', 6], ['A', 2]]  
Iterasi ke- 4 :  
    Proses A telah selesai diproses  
    Data proses yang tersisa : [['B', 6]]  
Iterasi ke- 5 :  
    Proses B sedang diproses, dan sisa waktu proses B = 3  
    Data proses yang tersisa : [['B', 3]]  
Iterasi ke- 6 :  
    Proses B telah selesai diproses  
    Data proses yang tersisa : []

---

(c)

Gambar 1: Penjadwalan dengan Struktur Data Queue