

# ITB\_IF2212\_2\_2324 Pemrograman Berioritasi Objek STI

[Dashboard](#) / [My courses](#) / [ITB\\_IF2212\\_2\\_2324](#) / [Praktikum 8: Collection and Nested Class](#) / [Praktikum 8](#)

Started on	Monday, 20 May 2024, 8:03 AM
State	Finished
Completed on	Monday, 20 May 2024, 9:43 AM
Time taken	1 hour 40 mins
Grade	100.00 out of 300.00 (33%)

Question **1**

Incorrect

Mark 0.00 out of 100.00

Flag question

Time limit	1 s
Memory limit	64 MB

Anda diminta untuk mengimplementasikan kelas `PenjurusanTPB`. Dalam kelas `PenjurusanTPB` terdapat kelas (Nested Class):

- Kelas Mahasiswa yang memiliki atribut NIM, Nama, IP, dan UKT. Terdapat konstruktor yang menerima NIM, Nama, IP, dan UKT. Terdapat pula aksesor untuk mengakses NIM, Nama, IP, dan UKT.
- Terdapat kelas `MahasiswaComparator` yang mengimplementasikan `Comparator<Mahasiswa>`. Kelas ini akan membandingkan antar-mahasiswa dengan prioritas sebagai berikut:
  - UKT yang lebih besar akan diprioritaskan.
  - IP yang lebih besar akan diprioritaskan.
  - NIM yang lebih kecil akan diprioritaskan. (NIM dipastikan unik)

Selain itu, terdapat method `PembangkitanAntrianPrioritas` yang menerima `List<Mahasiswa>` dan mengembalikan `PriorityQueue<Mahasiswa>` yang sudah terurut berdasarkan prioritas.

Lengkapilah file `'PenjurusanTPB.java'` dan kumpulkan kembali file `'PenjurusanTPB.java'` yang telah berisi jawaban Anda.

Docs:  
Priority Queue: <https://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html>  
Comparator: <https://docs.oracle.com/javase/8/docs/api/java/util/Comparator.html>

Java 8

PenjurusanTPB.java

Score: 0

Blackbox

Score: 0

Verdict: Compile error

Evaluator: Exact

No	Score	Verdict	Description
	0	Compile error	javac Main.java PenjurusanTPB.java PenjurusanTPB.java:52: error: illegal combination of modifiers: public and private public int compare(Mahasiswa mahasiswa1, Mahasiswa mahasiswa2) { ^ 1 error Makefile:28: recipe for target 'Main.class' failed make: *** [Main.class] Error 1

Question **2**

Incorrect

Mark 0.00 out of 100.00

Flag question

Time limit	1 s
Memory limit	64 MB

**Queue From Stack**

Anda diminta untuk mengimplementasikan Queue (First In First Out) dengan hanya menggunakan 2 stack. Queue yang diimplementasikan diperlukan untuk mendukung fungsi - fungsi dari sebuah queue yaitu push, peek, pop, dan empty.

void push(int x) : Menambah elemen x ke belakang Queue  
int pop() : Menghapus elemen dari depan Queue dan mengembalikan elemen tersebut  
int peek() : Mengembalikan elemen depan Queue  
boolean empty() : Mengembalikan apakah Queue kosong (true) atau tidak (false)

Anda hanya bisa menggunakan operasi standar dari sebuah stack yaitu:

- push(Integer item) : Menyimpan item ke puncak stack
- peek() : Melihat item yang berada di puncak stack tanpa mengeluarkannya dari stack
- pop() : Menghapus item di puncak stack dan mengembalikan item tersebut
- empty() : Menguji apakah stack kosong

Submit file `MyQueue.java`

Java 8

MyQueue.java

Score: 0

Blackbox

Score: 0

Verdict: Compile error

Evaluator: Exact

No	Score	Verdict	Description
	0	Compile error	javac Main.java MyQueue.java MyQueue.java:19: error: cannot find symbol s2.reversed(); ^ symbol: method reversed() location: variable s2 of type Stack<Integer> MyQueue.java:34: error: cannot find symbol s1.reversed(); ^ symbol: method reversed() location: variable s1 of type Stack<Integer> MyQueue.java:36: error: cannot find symbol s1.reversed(); ^ symbol: method reversed() location: variable s1 of type Stack<Integer> 3 errors make: *** [Main.class] Error 1 Makefile:27: recipe for target 'Main.class' failed

Question **3**

Correct

Mark 100.00 out of 100.00

Flag question

Time limit	1 s
Memory limit	64 MB

**Anagram**

Sebuah anagram adalah kata yang bisa dibentuk dengan mengubah urutan huruf-huruf pada sebuah kata. Sebagai contoh, kata "kasur" bisa diubah urutannya menjadi "rusak", kata "bisa" bisa diubah urutannya menjadi "sabi".

Dari contoh tersebut, bisa disimpulkan bahwa kata "kasur" merupakan anagram dari "rusak" dan kata "bisa" adalah anagram dari "sabi".

Pada soal kali ini, buatlah sebuah program `Main.java` yang memenuhi spesifikasi berikut:

- Program menerima input berikut:
  - Pada baris pertama, berisi bilangan bulat  $N$  ( $1 \leq N \leq 100$ ) menyatakan banyak kata yang akan diterima program.
  - Baris berikutnya berisi  $N$  buah kata yang dipisahkan spasi. Masing-masing kata hanya mengandung huruf kecil alfabet dan dijamin memiliki panjang  $\leq 50$ .
- Program mengeluarkan output berupa banyaknya anagram unik dari seluruh kata yang diterima.

**Contoh Input**

```
....
4
kasur bisa sabi rusak
....
```

**Contoh Output**

```
....
2
....
```

Penjelasan

Himpunan {"kasur", "rusak"} memiliki anagram yang sama, begitu juga dengan himpunan {"bisa", "sabi"}. Sehingga dari seluruh kata yang diterima, didapat 2 anagram unik.

Submit file **Main.java**.

**Hints:**  
Manfaatkan pengurutan *sorting* terhadap karakter pada kata untuk menentukan apakah kata satu dengan yang lainnya memiliki anagram yang sama.

Java 8

Main.java

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	5	Accepted	0.55 sec, 27.91 MB
2	5	Accepted	0.16 sec, 28.08 MB
3	5	Accepted	0.12 sec, 28.43 MB
4	5	Accepted	0.13 sec, 28.98 MB
5	5	Accepted	0.08 sec, 28.78 MB
6	5	Accepted	0.08 sec, 28.17 MB
7	5	Accepted	0.09 sec, 28.04 MB
8	5	Accepted	0.08 sec, 30.28 MB
9	5	Accepted	0.08 sec, 28.24 MB
10	5	Accepted	0.29 sec, 29.35 MB
11	10	Accepted	0.24 sec, 28.76 MB
12	10	Accepted	0.48 sec, 28.46 MB
13	10	Accepted	0.33 sec, 28.02 MB
14	10	Accepted	0.28 sec, 27.98 MB
15	10	Accepted	0.34 sec, 29.14 MB

[Finish review](#)

[← Praktikum 7 - Latihan](#)

Jump to...

[Praktikum 8 - Latihan →](#)