

Started on	Sunday, 27 April 2025, 7:12 PM
State	Finished
Completed on	Sunday, 27 April 2025, 9:08 PM
Time taken	1 hour 56 mins
Grade	300.00 out of 300.00 (100%)

Question **1**
Correct
Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Deque


Deque adalah sebuah struktur data *queue* yang memungkinkan kalian untuk melakukan *enqueue* dan *dequeue*. Deque merupakan singkatan dari *Double Ended Queue*

Berbeda dengan queue biasa yang hanya mendukung operasi di salah satu ujung (biasanya di belakang untuk enqueue dan di depan untuk dequeue), Deque memberikan fleksibilitas penuh dalam pengelolaan data. Struktur ini sangat berguna dalam berbagai kasus seperti:

- Implementasi antrian dengan prioritas fleksibel
- Penyimpanan data dalam urutan tertentu yang bisa berubah
- Algoritma sliding window
- Pemeriksaan palindrome

Disediakan berkas [Deque.java](#), lengkapi berkas tersebut dan kumpulkan berkas tersebut dengan nama **Deque.java**

Java 8 ▾

 [Deque.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.06 sec, 28.04 MB
2	20	Accepted	0.06 sec, 28.37 MB
3	20	Accepted	0.06 sec, 29.02 MB
4	20	Accepted	0.06 sec, 28.10 MB
5	20	Accepted	0.06 sec, 28.98 MB

Time limit	1 s
Memory limit	64 MB

Implementasikan kelas KeyValuePairGeneric

Key Value Pair Generi adalah sebuah kelas yang menyimpan tipe data generic berupa pair K,V dengan k merupakan key dan v merupakan value

Kelas tersebut memiliki variabel private berupa

- 1. key yang bertipe K dan menyimpan nilai key
- 2. value yang bertipe V dan menyimpan nilai value

Lengkapi metode dan kelas yang ada pada [KeyValuePairGeneric.java](#) Hint:

- 1. gunakan metode getClass().getSimpleName() untuk mencari tipe data

Java 8 ▾

 [KeyValuePairGeneric.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.29 sec, 27.94 MB
2	25	Accepted	0.34 sec, 26.27 MB
3	25	Accepted	0.37 sec, 28.08 MB
4	25	Accepted	0.36 sec, 28.61 MB

Time limit	1 s
Memory limit	64 MB

Generic - HashMap

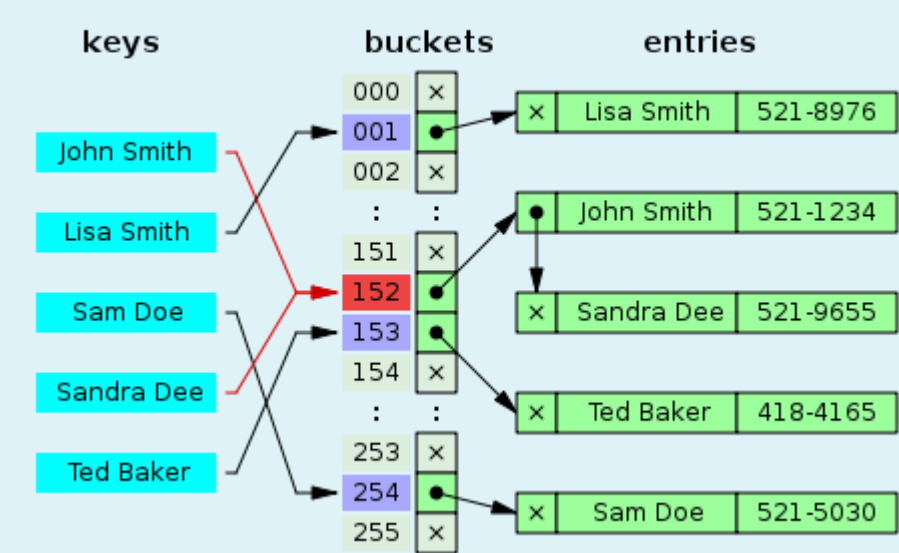
Map adalah sebuah struktur data yang menyimpan data dalam bentuk pasangan *key-value* unik. **Key** pada *map* merupakan identitas unik dari sebuah **Value** dan digunakan untuk menentukan lokasi penyimpanan dari *key-value*.

Salah satu contoh penggunaan dari struktur data *map* adalah untuk menyimpan nim dari mahasiswa. Misalkan terdapat *map* yang menyimpan pasangan *key-value* berupa **nama** sebagai *key* dan **nim** sebagai *value* seperti berikut.

```
map["Vincentpelit"] = "18221100"
map["Aulawesomegirl"] = "18221066"
map["sherra"] = "18221119"
```

Jika struktur data *map* bisa diakses dengan metode `get`, pemanggilan `map.get("Vincentpelit")` akan mengembalikan `"18221100"`, tetapi `map.get("Sapiqqqq")` akan mengembalikan `null` (karena *map* tidak menyimpan *key* = `"Sapiqqqq"`).

Hash Map merupakan struktur data *map* yang diimplementasikan dengan menggunakan nilai **hash** dari **Key** untuk menentukan lokasi penyimpanan. Lokasi penyimpanan dari pasangan *key-value* di dalam *hash map* disebut dengan **buckets**. Apabila terdapat beberapa pasangan *key-value* dengan **Key** yang berbeda dengan nilai *hash* yang sama, maka pasangan ini akan diletakan pada **buckets** yang sama sebagai **Linked List**. Ilustrasi dari *hash map* bisa dilihat pada gambar berikut.



sumber gambar: Wikipedia

Penjelasan:

Pada gambar, struktur data *hash map* menyimpan banyak pasangan *key-value* yang salah satunya adalah pasangan yang bernilai (John Smith, 521-1234). Pada pasangan dengan *key-value* tersebut, lokasi penyimpanan didapatkan dengan melakukan *hashing* pada John Smith, sehingga didapatkan lokasi penyimpanan pada *buckets* dengan index 152.

Tetapi pada salah satu pasangan lain, yaitu (Sandra Dee, 521-9655), hasil *hashing* dari Sandra Dee juga bernilai 152. Sehingga, hal ini menyebabkan pasangan *key-value* (Sandra Dee, 521-9655) harus disimpan sebagai *entry* setelah pasangan *key-value* (John Smith, 521-1234).

Buatlah kelas **HashMap** yang merupakan struktur data *hash map* yang dapat menyimpan pasangan *key-value* dengan tipe data apapun. Manfaatkan struktur data **Linked List** yang diberikan di attachment.

Lengkapi file [HashMap.java](#) yang mengimplementasi file [Map.java](#).

Attachment pelengkap: file [LinkedList.java](#) dan [Node.java](#)

Coba kerjakan sendiri sebelum bertanya ke Mbah GPT :3

Submit file **HashMap.java**

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	11	Accepted	0.06 sec, 30.32 MB
2	11	Accepted	0.06 sec, 29.29 MB
3	11	Accepted	0.07 sec, 29.40 MB
4	11	Accepted	0.06 sec, 28.99 MB
5	11	Accepted	0.07 sec, 28.20 MB
6	11	Accepted	0.18 sec, 28.52 MB
7	11	Accepted	0.10 sec, 28.37 MB
8	11	Accepted	0.10 sec, 27.88 MB
9	12	Accepted	0.13 sec, 31.64 MB

[◀ Pasca Kuis 1](#)

Jump to...



[Praktikum 5 ▶](#)