

---

# Pengenalan Sequenced Collection



# Sequenced Collection

- Sequenced Collection adalah fitur terbaru di Java 21
- Ini adalah fitur improvement dari Java Collection yang diajukan melalui JEP 431
- <https://openjdk.org/jeps/431>



# Kenapa Butuh Sequenced Collection

- Permasalahan dengan Java Collection sekarang adalah, tidak memiliki fitur untuk mempermudah kita untuk melakukan operasi pada awal atau akhir collection
- Salah satu class yang memiliki operasi untuk diawal dan diakhir adalah Deque, namun collection yang lain tidak memiliki hal itu, sehingga untuk melakukan operasi di awal dan diakhir, agak sedikit tricky



# Operasi Collection

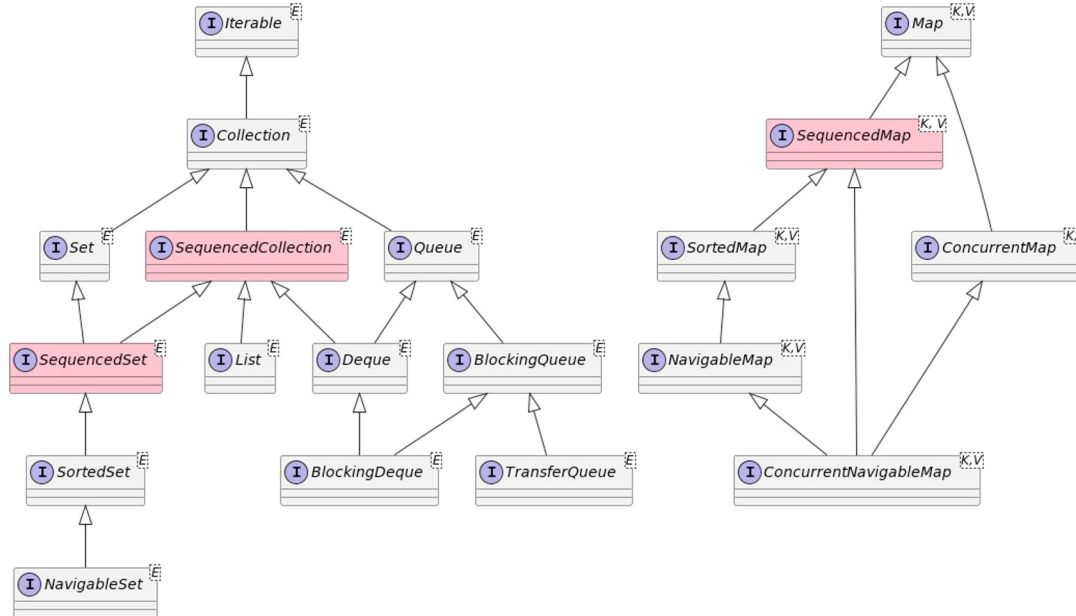
	First element	Last element
<b>List</b>	<code>list.get(0)</code>	<code>list.get(list.size() - 1)</code>
<b>Deque</b>	<code>deque.getFirst()</code>	<code>deque.getLast()</code>
<b>SortedSet</b>	<code>sortedSet.first()</code>	<code>sortedSet.last()</code>
<b>LinkedHashSet</b>	<code>linkedHashSet.iterator().next()</code>	<code>// missing</code>



# Sequenced Interface

- Sequenced Collection memperkenalkan 3 interface baru dalam Java Collection
- SequencedCollection,
- SequencedSet, dan
- SequencedMap

# Diagram : Sequenced Collection



---

# Membuat Project



# Membuat Project

- <https://start.spring.io/>



---

# Sequenced Collection



# Sequenced Collection

- Sequenced Collection merupakan parent interface untuk kebanyakan implementasi Class List, SortedSet dan Deque
- Terdapat banyak method yang bisa kita gunakan pada SequencedCollection, terutama untuk operasi diawal dan akhir data



# Sequenced Collection Method

```
▼ ⓘ 📁 SequencedCollection
  (m) 📁 reversed(): SequencedCollection<E>
  (m) 📁 addFirst(E): void
  (m) 📁 addLast(E): void
  (m) 📁 getFirst(): E
  (m) 📁 getLast(): E
  (m) 📁 removeFirst(): E
  (m) 📁 removeLast(): E
```



CollectionTest.java

```
SequencedCollection<String> list = new ArrayList<String>();  
list.addLast("Khannedy");  
list.addFirst("Kurniawan");  
list.addFirst("Eko");  
  
Assertions.assertEquals("Eko", list.getFirst());  
Assertions.assertEquals("Khannedy", list.getLast());  
  
SequencedCollection<String> reversed = list.reversed();  
Assertions.assertEquals(List.of("Khannedy", "Kurniawan", "Eko"), reversed);  
  
list.removeFirst();  
Assertions.assertEquals("Kurniawan", list.getFirst());  
  
list.removeLast();  
Assertions.assertEquals("Kurniawan", list.getLast());
```

---

# Sequenced Set



# Sequenced Set

- SequencedSet adalah interface untuk parent dari class implementasi dari SortedSet
- Kenapa interface Set bukan turunan dari SequencedSet? Karena dalam Set, tidak ada jaminan data akan berurut, oleh karena itu operasi awal dan akhir jadi tidak pasti



# Sequenced Set Method

```
✓ ⓘ SequencedSet  
  ⓘ reversed(): SequencedSet<E> ↑ SequencedCol
```



# Manipulasi Data

- Karena SortedSet merupakan struktur data yang dijamin berurut posisinya, maka secara otomatis operasi addFirst() dan addLast() tidak bisa dilakukan di SortedSet, hal ini karena walaupun kita simpan di awal, secara otomatis SortedSet akan mengurutkan sesuai posisi yang benar
- Oleh karena itu untuk menambah data, kita tetap menggunakan add()
- Sedangkan untuk removeFirst() dan removeLast(), itu bisa dilakukan





SetTest.java

```
SequencedSet<String> set = new TreeSet<>();  
set.add("Eko");  
set.add("Budi");  
set.add("Joko");  
  
Assertions.assertEquals("Budi", set.getFirst());  
Assertions.assertEquals("Joko", set.getLast());  
  
SequencedSet<String> reversed = set.reversed();  
Assertions.assertEquals("Budi", reversed.getLast());  
Assertions.assertEquals("Joko", reversed.getFirst());
```

---

# Sequenced Map



# Sequenced Map

- SequencedMap adalah interface untuk implementasi class SortedMap
- Map biasa bukanlah turunan dari SequencedMap, hal ini karena dalam Map biasa, tidak ada jaminan bahwa urutan key, sehingga operasi diawal dan diakhir tidak dimungkinkan
- Oleh karena itu, hanya turunan dari SortedMap yang bisa menggunakan SequencedMap

# Sequenced Map Method

## SequencedMap

- reversed(): SequencedMap<K, V>
- firstEntry(): Entry<K, V>
- lastEntry(): Entry<K, V>
- pollFirstEntry(): Entry<K, V>
- pollLastEntry(): Entry<K, V>
- putFirst(K, V): V
- putLast(K, V): V
- > sequencedKeySet(): SequencedSet<K>
- > sequencedValues(): SequencedCollection<V>
- > sequencedEntrySet(): SequencedSet<Entry<K, V>>



MapTest.java

```
SequencedMap<String, String> map = new TreeMap<>();  
map.put("c", "Eko");  
map.put("b", "Kurniawan");  
map.put("a", "Khannedy");  
  
Assertions.assertEquals("Khannedy", map.firstEntry().getValue());  
Assertions.assertEquals("Eko", map.lastEntry().getValue());  
  
SequencedMap<String, String> reversed = map.reversed();  
Assertions.assertEquals("Khannedy", reversed.lastEntry().getValue());  
Assertions.assertEquals("Eko", reversed.firstEntry().getValue());
```