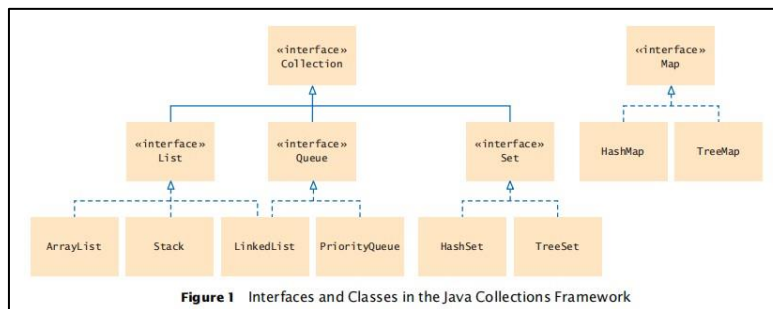
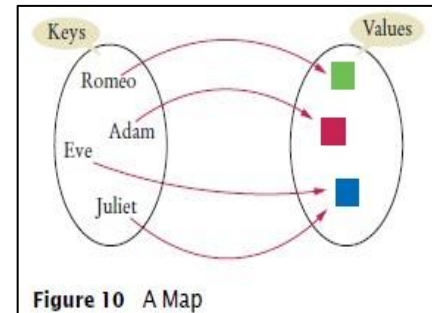
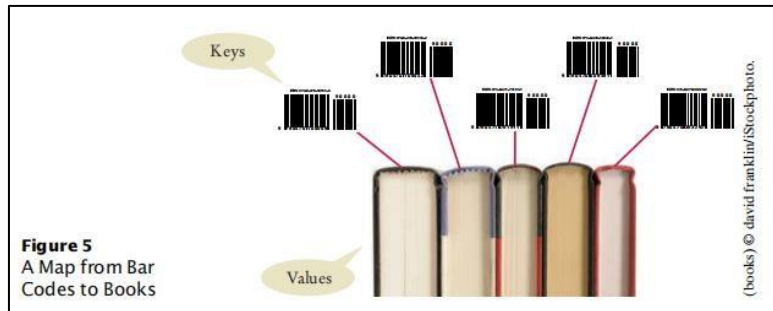


## [8] Map & HashMap

### Map



- Map merupakan interface yang tersedia di Java yang menyimpan data dalam pasangan *key* (kunci) dan *value* (nilai).
- Map tidak dapat dilakukan dengan *key* yang duplikat dan setiap *key* memuat hanya ke satu *value*.
- Map memungkinkan *retrieval* (pemanggilan), *deletion* (penghapusan), dan *updating* (pembaruan) secara cepat hanya melalui *key*.
- Nilai (*value*) dapat berupa bentuk string, integer, boolean, float, double, dan objek, sedangkan untuk *key* biasanya dalam bentuk string dan integer.
- Jika mencoba menambahkan elemen dengan *key* yang sama, maka elemen sebelumnya pada *key* tersebut akan ditimpa sehingga hilang.

### Hashmap

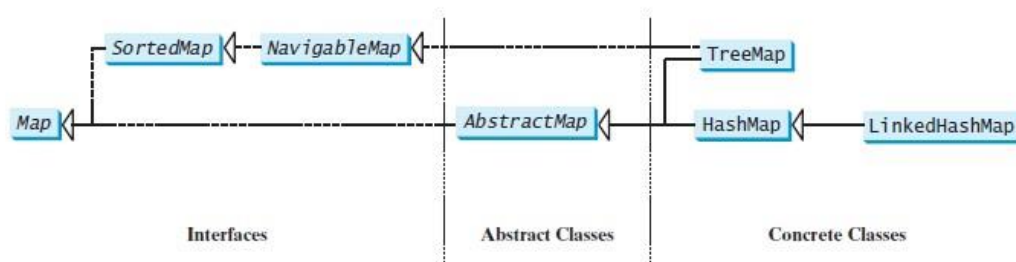


FIGURE 21.3 A map stores key/value pairs.

- Hashmap merupakan implementasi dari class Map.
- HashMap efisien untuk menemukan nilai, memasukkan nilai, dan menghapus nilai.

Perhatikan table berikut yang terdiri dari pasangan *key* dan *value*, seperti inilah isi dari class atau objek HashMap.

Key	Value
"name"	"Muhammad Muftahillah"
"id"	2411016210015
"major"	"Computer Science"
"isGraduated"	false

#### Contoh Program (menggunakan Java Collection)

```
import java.util.HashMap;

public class CobaHashMap {
    public static void main(String[] args) {
        // Membuat objek HashMap
        HashMap<Integer, String> days = new HashMap<Integer, String>();

        // Mengisi nilai ke objek days
        days.put(1, "Minggu");
        days.put(2, "Senin");
        days.put(3, "Selasa");
        days.put(4, "Rabu");
        days.put(5, "Kamis");
        days.put(6, "Jumat");
        days.put(7, "Sabtu");

        // Mencetak semua isi dari objek days
        System.out.println("Isi objek days: " + days);
    }
}
```

Berikut ini langkah pembuatan HashMap dan beberapa contoh implementasinya:

#### 1. Mengimport HashMap

```
import java.util.HashMap;
```

#### 2. Membuat objek HashMap

Diagram illustrating the declaration of a HashMap:

```
HashMap<K,V> hm = new HashMap<K,V>
```

Annotations:

- Tipe data key (points to K)
- Tipe data vlue (points to V)
- objek HashMap (points to new)

petanikode.com

Tipe data untuk key biasanya dalam bentuk integer dan string. Untuk tipe data value, bisa juga dalam bentuk class.

```
HashMap<Integer, String> days = new HashMap<Integer, String>();
```

### 3. Mengisi nilai

```
days.put(1, "Minggu");
days.put(2, "Senin");
days.put(3, "Selasa");
days.put(4, "Rabu");
days.put(5, "Kamis");
days.put(6, "Jumat");
days.put(7, "Sabtu");
```

### 4. Mengambil nilai

```
// mengambil hari senin
days.get(2)
```

### 5. Menghapus nilai

```
// menghapus hari senin
days.remove(2);
System.out.println("Isi objek days: " + days);

// menghapus semua hari days.clear();
System.out.println("Isi objek days: " + days);
```

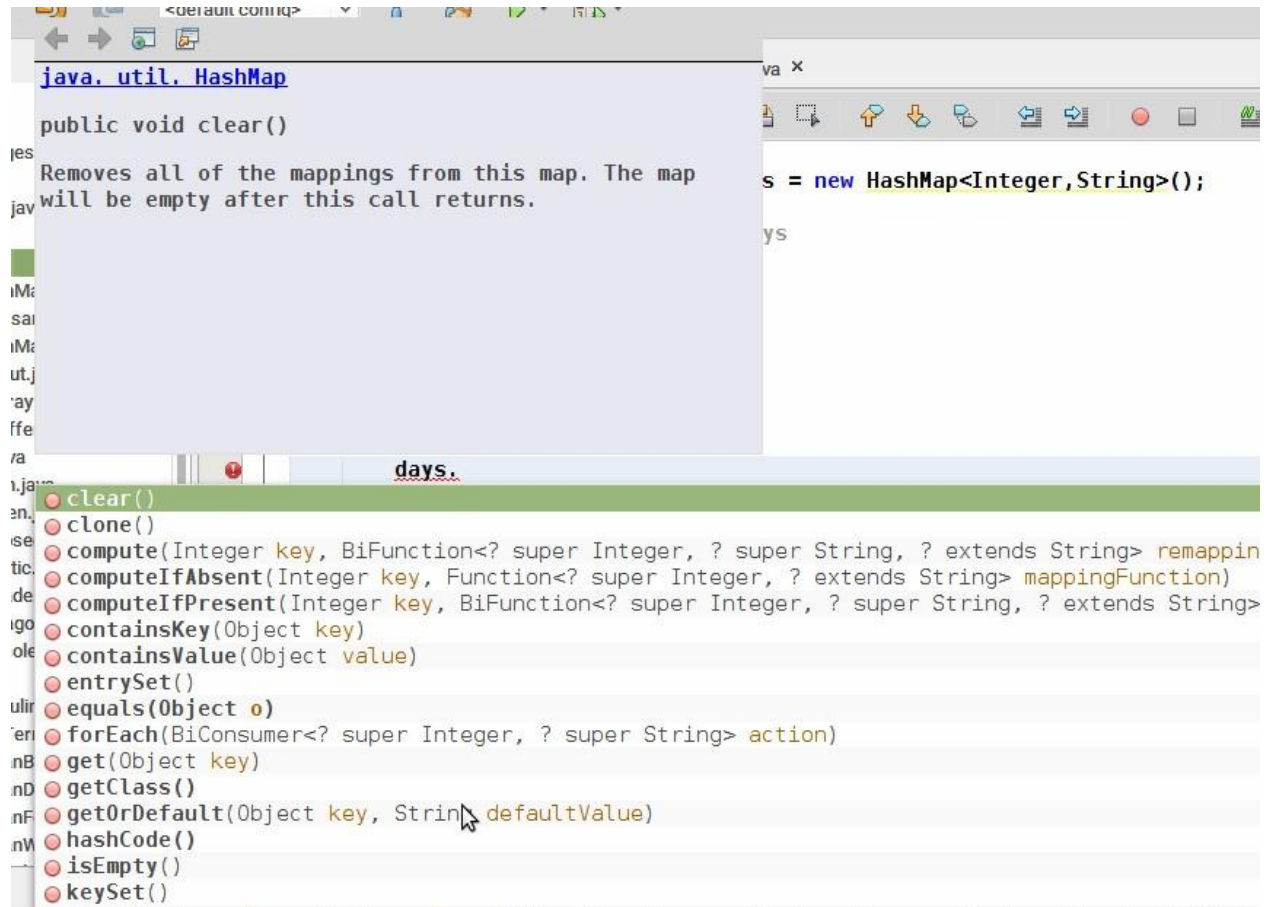
### 6. Mengubah nilai dan kunci dari HashMap

```
// mengubah hari minggu menjadi hari Ahad
days.put(1, "Ahad");
// mengubah hari rabu menjadi wednesday
days.replace(4, "wednesday");
```

Method `put()` dan `replace()` keduanya sama-sama bisa digunakan untuk mengubah nilai. Pada method `put()`, jika `key` yang diberikan sudah ada dalam `HashMap`, maka `put()` akan menggantikan nilai lama yang terkait dengan `key` tersebut dengan nilai baru yang diberikan. Jika `key` belum ada dalam `HashMap`, `put()` akan menambahkan `key-value` baru ke dalam `HashMap`. Pada method `replace`, jika `key` yang diberikan sudah ada di dalam `HashMap`, `replace()` akan menggantikan nilai lama dengan nilai baru

yang diberikan. Jika key tidak ada di dalam HashMap, `replace()` tidak akan menambahkan key-value baru dan tidak akan mengubah HashMap.

### Method pada class HashMap



Sebenarnya masih banyak lagi method HashMap lain yang dapat kita coba. Berikut ini penjelasan beberapa method:

- `clear()` untuk membersihkan isi HashMap;
- `isEmpty()` untuk mengecek apakah HashMap dalam keadaan kosong atau tidak;
- `size()` untuk mengambil ukuran HashMap (jumlah item di dalam hashmap);
- `values()` untuk mengambil semua nilai yang ada di dalam HashMap;
- `keySet()` untuk mengambil semua key yang ada di dalam HashMap;
- `clone()` untuk menggandakan objek HashMap;
- dll.

### Contoh Program

Buku.java

```

public class Buku {
    private String title;
    private String author;

    public Buku(String title, String author) {
        this.title = title;
        this.author = author;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }
}

```

#### BukuHashMap.java

```

import java.util.HashMap;
import java.util.Map;

public class BukuHashMap {
    public static void main(String[] args) {

        // membuat objek hashmap
        HashMap<String, Buku> books = new HashMap<String, Buku>();

        // membuat objek buku
        Buku bukuJava = new Buku("Tutorial Java", "Petani Kode");
        Buku bukuKotlin = new Buku("Pemrograman Kotlin", "Petani Kode");
        Buku bukuAndroid = new Buku("Pemrograman Android", "Petani Kode");

        // mengisi objek hashmap dengan objek buku
        books.put("java", bukuJava);
        books.put("kotlin", bukuKotlin);
        books.put("android", bukuAndroid);

        // cetak semua buku
        for (Map.Entry<String, Buku> entry : books.entrySet()) {
            Buku buku = entry.getValue();
            System.out.println(entry.getKey() + ": " + buku.getTitle());
        }
    }
}

```