

# Mengenal HashMap dalam Pemrograman Java

---

Oleh:

Taryana Suryana M.Kom  
Teknik Informatika Unikom

[taryanarx@email.unikom.ac.id](mailto:taryanarx@email.unikom.ac.id)

[taryanarx@gmail.com](mailto:taryanarx@gmail.com)

[Line/Telegram: 081221480577](https://t.me/081221480577)

Join Zoom Meeting

<https://us04web.zoom.us/j/3629329963?pwd=ZUhWcXl6RHp3dTNKZmlxWkUrV2ZCQT09>

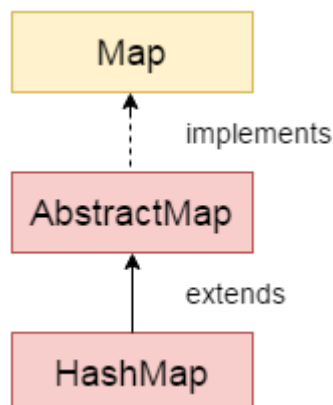
Meeting ID: 362 932 9963

Passcode: 8HQiXn

Class **HashMap** merupakan class turunan dari class **AbstractMap** dan implementasi dari interface **Map**.

HashMap Merupakan sebuah class yang berisi sekumpulan pasangan nilai (*value*) dan kunci (*key*).

Nilai bisa dalam bentuk string, integer, boolean, float, double, dan objek. Sedangkan untuk *key* biasanya dalam bentuk string dan integer.



Coba perhatikan tabel berikut:

| KEY     | VALUE                  |
|---------|------------------------|
| "Nama"  | "Fahra Ragita"         |
| "Email" | "fahra@gmail.com"      |
| "1"     | "Senin"                |
| "url"   | "https://ciwaruga.com" |

Tabel di atas terdiri dari pasangan *key* dan *value*, seperti inilah isi dari class atau objek HashMap.

## Membuat HashMap

Sebelum bisa menggunakan HashMap, kita harus mengimpornya terlebih dahulu:

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

Setelah itu baru kita bisa menggunakannya.

Nah!, untuk menggunakan HashMap, kita harus membuat objeknya terlebih dahulu. Objek dari Hashmap dapat dibuat dengan kata kunci *new*.

Namun, ada sedikit tambahan untuk menentukan tipe data untuk key dan value yang akan disimpan.

```
HashMap<Key, Value> hm = new HashMap<Key, Value>
```

Tipe data Key

Tipe data value

Objek HashMap

*Tipe data untuk key biasanya Integer atau String. Sedangkan untuk Value selain Integer atau String dapat juga dalam bentuk Class.*

Contoh:

```
HashMap<String, String> IbuKota = new HashMap<String, String>
```

Pada contoh di atas, kita membuat objek HashMap bernama IbuKota. Objek ini dapat kita gunakan untuk menyimpan koleksi data.

Tipe data yang digunakan untuk *key* adalah String dan *value* adalah *String*.

**Artinya:** key harus bertipe String dan nilai yang tersimpan harus dalam bertipe String. Untuk lebih jelasnya, silahkan lanjutkan...

## Mengisi Nilai ke dalam HashMap

---

Tadi kita sudah membuat objek hashmap bernama IbuKota dengan tipe data:

- K (key): String
- V (value): String

Untuk mengisi nilai kedalam objek IbuKota, dapat menggunakan method PUT, perhatikan contoh berikut:

```
IbuKota.put("Jawa Barat", "Bandung");
IbuKota.put("Jawa Tengan", "Semarang");
IbuKota.put("Jawa Timur", "Surabaya");
IbuKota.put("DKI", "Jakarta");
```

Objek HashMap **IbuKota** berisi nama-nama Ibu Kota Provinsi yang ada di Indonesia dengan Key Nama Provinsi.

```
package hashmap;
import java.util.HashMap;
public class Hashmap {
    public static void main(String[] args) {
        HashMap<String, String> IbuKota = new HashMap<String, String>();
        IbuKota.put("Jawa Barat", "Bandung");
        IbuKota.put("Jawa Tengan", "Semarang");
        IbuKota.put("Jawa Timur", "Surabaya");
        IbuKota.put("DKI", "Jakarta");
        System.out.println(IbuKota);
    }
}
```

run:

```
{DKI=Jakarta, Jawa Tengan=Semarang, Jawa Barat=Bandung, Jawa Timur=Surabaya}
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Mengambil Nilai dari HashMap

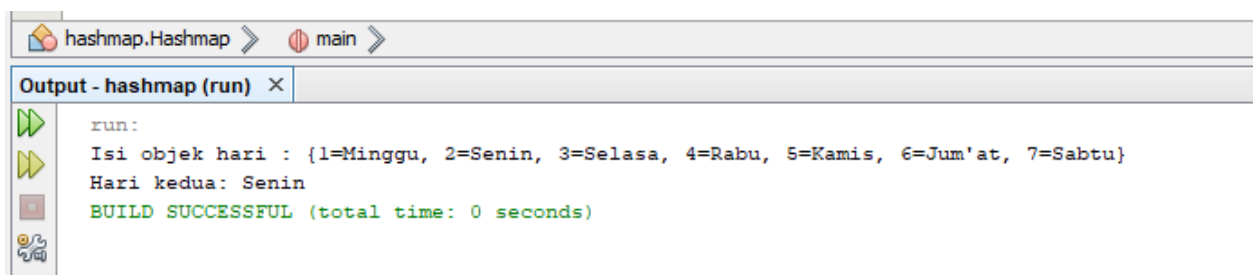
Untuk mengambil nilai dari HashMap, kita bisa menggunakan method `get()` dengan parameter kuncinya.

Contoh:

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

```
package hashmap;
import java.util.HashMap;
public class Hashmap {
    public static void main(String[] args) {
        // TODO code application logic here
        HashMap<Integer, String> hari = new HashMap<Integer, String>();
        // mengisi nilai ke objek Hari
        hari.put(1, "Minggu");
        hari.put(2, "Senin");
        hari.put(3, "Selasa");
        hari.put(4, "Rabu");
        hari.put(5, "Kamis");
        hari.put(6, "Jum'at");
        hari.put(7, "Sabtu");

        // mencetak semua isi dari objek hari
        System.out.println("Isi objek hari : " + hari);
        System.out.println("Hari kedua: " + hari.get(2));
    }
}
```



```
run:
Isi objek hari : {1=Minggu, 2=Senin, 3=Selasa, 4=Rabu, 5=Kamis, 6=Jum'at, 7=Sabtu}
Hari kedua: Senin
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Menghapus Nilai dari HashMap

Ada dua method yang dapat digunakan untuk menghapus nilai dari HashMap:

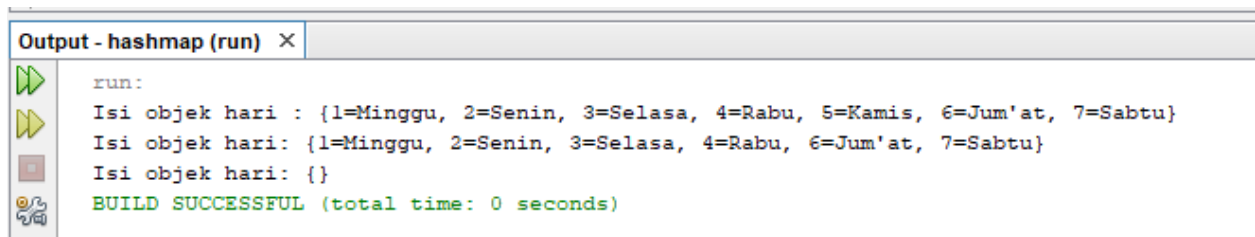
1. `remove()` menghapus salah satu nilai;
2. `clear()` menghapus semua nilai;

```
package hashmap;
import java.util.HashMap;
public class Hashmap {
    public static void main(String[] args) {
        // TODO code application logic here
        HashMap<Integer, String> hari = new HashMap<Integer,String>();
        // mengisi nilai ke objek Hari
        hari.put(1, "Minggu");
        hari.put(2, "Senin");
        hari.put(3, "Selasa");
        hari.put(4, "Rabu");
        hari.put(5, "Kamis");
        hari.put(6, "Jum'at");
        hari.put(7, "Sabtu");

        // mencetak semua isi dari objek hari
        System.out.println("Isi objek hari : " + hari);

        // menghapus malam Jumat <-- Duda Merana detected :-)
        hari.remove(5);
        System.out.println("Isi objek hari: " + hari);

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



```
Output - hashmap (run) X
run:
Isi objek hari : {1=Minggu, 2=Senin, 3=Selasa, 4=Rabu, 5=Kamis, 6=Jum'at, 7=Sabtu}
Isi objek hari: {1=Minggu, 2=Senin, 3=Selasa, 4=Rabu, 6=Jum'at, 7=Sabtu}
Isi objek hari: {}
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Mengubah Nilai dan Kunci dari HashMap

---

Ada dua method yang dapat digunakan untuk mengubah nilai di dalam HashMap:

1. Method `put()`
2. Method `replace()`

Apa bedanya?

Saya kira tidak ada perbedaan. Keduanya sama-sama bisa digunakan untuk mengubah nilai.

Namun, sepertinya untuk method `replace()` .....nilai yang akan diubah harus sudah ada di dalam HashMap.

Sedangkan untuk method `put()`, dia akan menambahkan yang baru apabila belum ada di dalam HashMap.

```
package hashmap;
import java.util.HashMap;
public class Hashmap {

    public static void main(String[] args) {
        // TODO code application logic here
        HashMap<Integer, String> hari = new HashMap<Integer,String>();
        // mengisi nilai ke objek Hari
        hari.put(1, "Minggu");
        hari.put(2, "Senin");
        hari.put(3, "Selasa");
        hari.put(4, "Rabu");
        hari.put(5, "Kamis");
        hari.put(6, "Jum'at");
        hari.put(7, "Sabtu");

        // mencetak semua isi dari objek hari
        System.out.println("Isi objek hari : " + hari);
        hari.put(1, "Ahad");

        // mengubah hari Jumat menjadi Jumaah
        hari.replace(6, "Jumaah");
        System.out.println("Isi objek hari: " + hari);
    }
}
```

```
Output - hashmap (run) ×
run:
Isi objek hari : {1=Minggu, 2=Senin, 3=Selasa, 4=Rabu, 5=Kamis, 6=Jum'at, 7=Sabtu}
Isi objek hari: {1=Ahad, 2=Senin, 3=Selasa, 4=Rabu, 5=Kamis, 6=Jumaah, 7=Sabtu}
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Ukuran HashMap

Untuk mengetahui berapa banyak item yang ada dalam HashMap, gunakan Metode `size()`:

```
hari.size()); //7
```

## Loop Melalui HashMap

MEnampilkan seluruh item dari sebuah HashMap, dapat dilakukan dengan menggunakan perintah perulangan **for-each** .

**Catatan:** Gunakan Metode `keySet()` jika Anda hanya menginginkan kunci, dan gunakan Metode `values()` jika Anda hanya menginginkan nilai:

```
import java.util.HashMap;

public class Main {
    public static void main(String[] args) {
        HashMap<Integer, String> hari = new HashMap<Integer, String>();
        // mengisi nilai ke objek Hari
        hari.put(1, "Minggu");
        hari.put(2, "Senin");
        hari.put(3, "Selasa");
        hari.put(4, "Rabu");
        hari.put(5, "Kamis");
        hari.put(6, "Jum'at");
        hari.put(7, "Sabtu");

        //menampilkan key
        for (Integer i : hari.keySet()) {
            System.out.println(i);
        }

        //menampilkan value
        for (String i : hari.values()) {
            System.out.println(i);
        }
    }
}
```

```
}  
  
}  
}
```

Hasil Run

```
1  
2  
3  
4  
5  
6  
7  
Minggu  
Senin  
Selasa  
Rabu  
Kamis  
Jum'at  
Sabtu
```

## Menampilkan key dan value

---

Untuk menampilkan key dan value HashMap dapat dilakukan dengan cara seperti berikut:

```
for (Integer i : hari.keySet()) {  
    System.out.println("key: " + i + " value: " + hari.get(i));  
}
```

Hasil Run:

```
key: 1 value: Minggu  
key: 2 value: Senin  
key: 3 value: Selasa  
key: 4 value: Rabu  
key: 5 value: Kamis  
key: 6 value: Jum'at  
key: 7 value: Sabtu
```



## Tipe yang lain

Key dan Value dalam HashMap sebenarnya adalah objek. Dalam contoh di atas, kami menggunakan objek bertipe "String". Ingat bahwa String di Java adalah objek (bukan tipe primitif). Untuk menggunakan tipe lain, seperti int, Anda harus menentukan [kelas pembungkus yang](#) setara : **Integer**. Untuk tipe primitif lainnya, gunakan: **Boolean** untuk boolean, **Character** untuk char, **Double** untuk double, dll:

## Contoh

Buat Objek **HashMap** bernama **orang** yang akan menyimpan Kunci **String** dan Nilai **Integer**

```
// Import the HashMap class
import java.util.HashMap;

public class Main {
    public static void main(String[] args) {
        //Buat sebuah objek HashMap namain orang;
        HashMap<String, Integer> orang = new HashMap<String, Integer>();

        // Key dan Value (Nama, Umur)
        orang.put("Fahra", 20);
        orang.put("Gaga", 6);
        orang.put("Dewi", 17);
        orang.put("Ilham", 25);

        for (String i : orang.keySet()) {
            System.out.println("Nama: " + i + " Umur: " + orang.get(i));
        }
    }
}
```

Hasil Run:

```
Nama: Dewi Umur: 17
Nama: Fahra Umur: 20
Nama: Gaga Umur: 6
Nama: Ilham Umur: 25
```

## DAFTAR PUSTAKA

1. Suryana, Taryana (2021) [Mengetahui Hashset Dalam Pemrograman Java](#). [Teaching Resource]
2. Suryana, Taryana (2021) [Pembuatan Fungsi Dalam Bahasa Pemrograman Dart](#). [Teaching Resource]
3. Suryana, Taryana (2021) [Kelas Dan Objek Dalam Pemrograman Dart](#). [Teaching Resource]
4. Suryana, Taryana (2021) [Fungsi Logika Atau Percabangan Dalam Dart](#). [Teaching Resource]
5. Suryana, Taryana (2021) [Belajar Bahasa Pemrograman Dart](#). [Teaching Resource]
6. Suryana, Taryana (2021) [Input Dan Output Dalam Dart](#). [Teaching Resource]
7. Suryana, Taryana (2021) [Variabel Dan Tipe Data Dalam Dart](#). [Teaching Resource]
8. Suryana, Taryana (2021) [Operator Dalam Bahasa Pemrograman Dart](#). [Teaching Resource]
9. Suryana, Taryana (2021) [List Dan Map Dalam Pemrograman Dart](#). [Teaching Resource]
10. Suryana, Taryana (2021) [Perulangan Dalam Pemrograman Dart](#). [Teaching Resource]
11. Suryana, Taryana; Koesharyatin (2014), [Aplikasi Internet Menggunakan HTML, CSS & Java Script](#), Elexmedia Komputindo, Jakarta
12. Suryana, Taryana (2021) [Bab 1. Pengenalan Html](#). [Teaching Resource]
13. Suryana, Taryana (2021) [Pelatihan Pembuatan Website Dengan Menggunakan Html Dan Dreamweaver](#). [Teaching Resource]
14. Suryana, Taryana (2010) [Membuat Web Pribadi dan Bisnis dengan HTML](#), Gava Media, Yogyakarta
15. Suryana, Taryana (2021) [Membuat Form Input Dan Output Java Menggunakan Netbeans](#). [Teaching Resource]
16. Suryana, Taryana (2021) [Memulai Menggunakan Apache Netbeans 12](#). [Teaching Resource]
17. Suryana, Taryana (2021) [Menggunakan Array Dalam Java](#). [Teaching Resource]
18. Suryana, Taryana (2021) [Membuat Form Input Dan Output Java Menggunakan Netbeans](#). [Teaching Resource]
19. Suryana, Taryana; Koesharyatin (2014), [Aplikasi Internet Menggunakan HTML, CSS & Java Script](#), Elexmedia Komputindo, Jakarta
20. Suryana, Taryana (2021) [Bab 1. Pengenalan Html](#). [Teaching Resource]
21. Suryana, Taryana (2021) [Pelatihan Pembuatan Website Dengan Menggunakan Html Dan Dreamweaver](#). [Teaching Resource]
22. Suryana, Taryana (2010) [Membuat Web Pribadi dan Bisnis dengan HTML](#), Gava Media, Yogyakarta
23. <https://www.w3schools.com/html>, September, 2021
24. <https://www.w3schools.com/java>
25. <https://petanikode.com>