

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : COLLECTION
 NAMA : Dani Adrian
 NIM : 225150201111009
 ASISTEN : Tengku Muhammad Rafi Rahardiansyah
 Muhammad Bin Djafar Almasyhur
 TGL PRAKTIKUM : 17 Mei 2023

A. Collection

Pelaksanaan Percobaan

Set

Pertanyaan

1. Ketikkan kode program dibawah ini dan analisis output dari program tersebut!

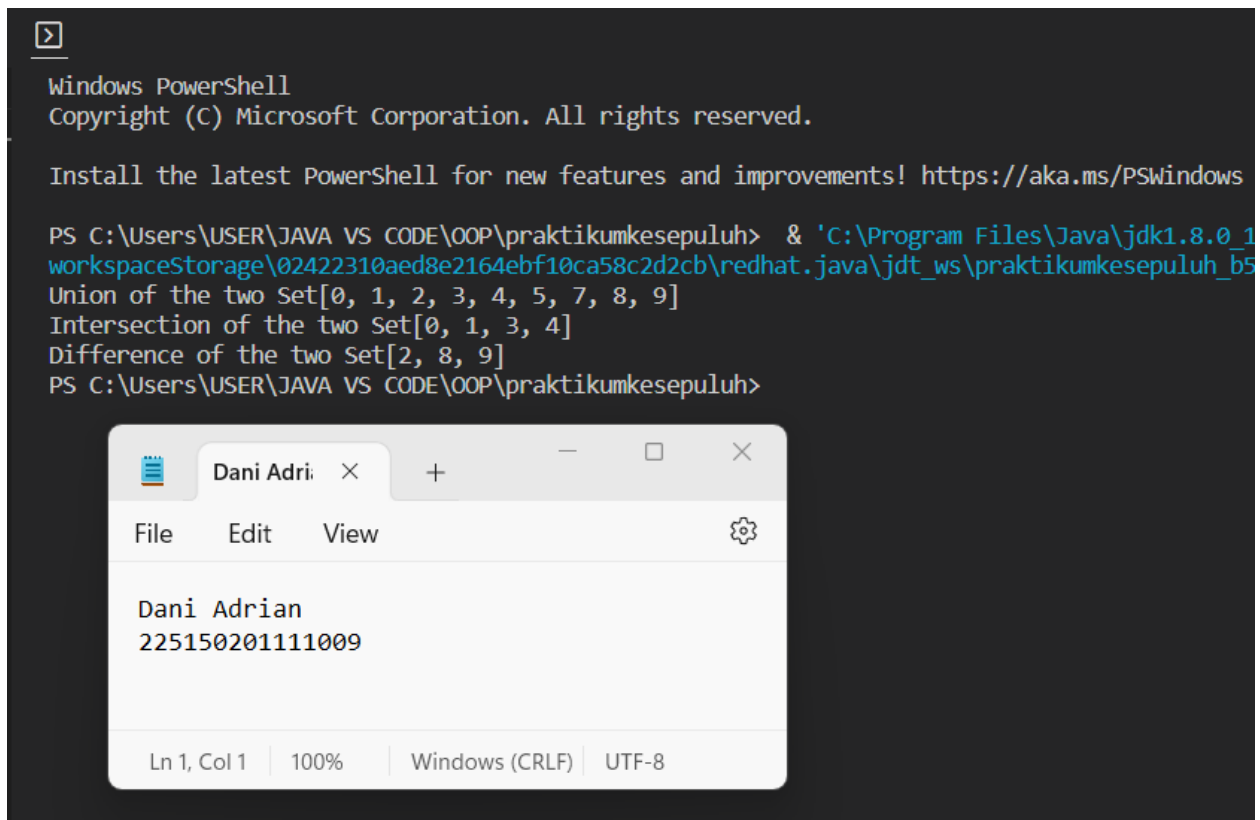
SetExample.java	
1.	// Java Program Demonstrating Operations on the Set
2.	// such as Union. Intersection and Difference operations
3.	
4.	//Importing all utility classes
5.	import java.util.*;
6.	
7.	//Main class
8.	public class SetExample {
9.	
10.	// Main driver method
11.	public static void main(String args[]) {
12.	// Creating an object of Set class
13.	// Declaring object of Integer type
14.	Set<Integer> a = new HashSet<Integer>();
15.	
16.	// Adding all elements to list
17.	a.addAll(Arrays.asList(new Integer[] { 1, 3, 2,
18.	4, 8, 9, 0 }));
19.	// Again declaring object of Set class
20.	// with reference to HashSet
21.	Set<Integer> b = new HashSet<Integer>();
22.	b.addAll(Arrays.asList(new Integer[] { 1, 3, 7,
23.	5, 4, 0, 7, 5 }));
24.	// To find union
25.	Set<Integer> union = new HashSet<Integer>(a);
26.	union.addAll(b);

27.	System.out.print("Union of the two Set");
28.	System.out.println(union);
29.	
30.	// To find intersection
31.	Set<Integer> intersection = new
	HashSet<Integer>(a);
32.	intersection.retainAll(b);
33.	System.out.print("Intersection of the two Set");
34.	System.out.println(intersection);
35.	
36.	// To find the symmetric difference
37.	Set<Integer> difference = new
	HashSet<Integer>(a);
38.	difference.removeAll(b);
39.	System.out.print("Difference of the two Set");
40.	System.out.println(difference);
41.	}
42.	}

Source Code

1.	Tulis source code di sini pake courier new 12
----	---

Output



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> & 'C:\Program Files\Java\jdk1.8.0_1
workspaceStorage\02422310aed8e2164ebf10ca58c2d2cb\redhat.java\jdt_ws\praktikumkesepuluh_b5
Union of the two Set[0, 1, 2, 3, 4, 5, 7, 8, 9]
Intersection of the two Set[0, 1, 3, 4]
Difference of the two Set[2, 8, 9]
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>

```

Dani Adrian
225150201111009

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Penjelasan

- “a” : Set<Integer>
- “b” : HashSet<Integer>
- Objek "a" : {1, 3, 2, 4, 8, 9, 0}
- Objek "b" : {1, 3, 7, 5, 4, 0, 7, 5}.

Union

Program membuat objek baru "union" sebagai salinan dari objek "a" menggunakan konstruktor HashSet<Integer>(a). Selanjutnya, elemen-elemen objek "b" ditambahkan ke objek "union" menggunakan metode addAll().

Hasil union selanjutnya dicetak, hasil union: [0, 1, 2, 3, 4, 5, 7, 8, 9]

Intersection

Program membuat objek baru "intersection" sebagai salinan dari objek "a" menggunakan konstruktor HashSet<Integer>(a). Kemudian, hanya elemen-elemen yang terdapat di objek "b" yang dipertahankan dalam objek "intersection" menggunakan metode retainAll().

Hasil intersection selanjutnya dicetak. hasil intersection: [0, 1, 3, 4]

Difference

Program membuat objek baru "difference" sebagai salinan dari objek "a" menggunakan konstruktor `HashSet<Integer>(a)`. Kemudian, elemen-elemen yang terdapat di objek "b" dihapus dari objek "difference" menggunakan metode `removeAll()`.

Hasil selanjutnya dicetak, hasil difference: [2, 8, 9]

Map

Pertanyaan

1. Ketikkan kode program dibawah ini dan analisis output dari program tersebut!

MapExample.java	
1.	// Java program to demonstrate
2.	// the working of Map interface
3.	
4.	import java.util.*;
5.	class MapExample {
6.	public static void main(String args[])
7.	{
8.	// Initialization of a Map
9.	// using Generics
10.	Map<Integer, String> hm1 = new HashMap<Integer,
	String>();
11.	
12.	// Inserting the Elements
13.	hm1.put(1, "Geeks");
14.	hm1.put(2, "Geeks");
15.	hm1.put(3, "Geeks");
16.	
17.	System.out.println("Initial Map: " + hm1);
18.	
19.	hm1.put(new Integer(2), "For");
20.	
21.	System.out.println("Updated Map " + hm1);
22.	
23.	hm1.remove(new Integer(3));
24.	
25.	// Final Map

26.	<code>System.out.println("Final Map " + hm1);</code>
27.	<code>}</code>
28.	<code>}</code>

Source Code

1.	Tulis source code di sini pake courier new 12
----	---

Output

The screenshot shows a Windows command prompt window with the following text:

```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> cd 'c:\Us
in\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspac
e3\bin' 'MapExample'
Initial Map: {1=Geeks, 2=Geeks, 3=Geeks}
Updated Map {1=Geeks, 2=For, 3=Geeks}
Final Map {1=Geeks, 2=For}
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>
```

Below the command prompt is a Notepad window titled "Dani Adri". The text in the Notepad window is:

```
Dani Adrian
225150201111009
```

The Notepad window also shows a status bar at the bottom with the text: "Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8".

Penjelasan

Objek `Map<Integer, String>` yaitu "hm1" menggunakan kelas `HashMap<Integer, String>`.

Elemen-elemen dimasukkan ke dalam objek "hm1" menggunakan metode `put()`.

Tiga pasangan key-value :

- (1, "Geeks")
- (2, "Geeks")

- (3, "Geeks")

Isi dari objek "hm1" dicetak menggunakan metode `println()`. Menghasilkan output "Initial Map {1=Geeks, 2=Geeks, 3=Geeks}" yang menunjukkan elemen-elemen yang ditambahkan ke dalam map.

Elemen dengan key 2 diupdate menjadi "For" menggunakan metode `put()`.

Output "Updated Map {1=Geeks, 2=For, 3=Geeks}" mencerminkan perubahan tersebut.

Elemen dengan key 3 dihapus menggunakan metode `remove()`.

Output "Final Map {1=Geeks, 2=For}" menunjukkan map setelah elemen dengan key 3 dihapus.

B. Tugas Praktikum

Apabila diketahui data anggota tim futsal sebagai berikut :

No	Tim A		TIM B	
	Tinggi Badan (cm)	Berat Badan (kg)	Tinggi Badan (cm)	Berat Badan (kg)
1.	168	50	170	66
2.	170	60	167	60
3.	165	56	165	59
4.	168	55	166	58
5.	172	60	168	58
6.	170	70	175	71
7.	169	66	172	68
8.	165	56	171	68
9.	171	72	168	65
10.	166	56	169	60

Pertanyaan

1. Dengan program java, carilah data pemain diantara kedua tim tersebut :
 - a. Yang mempunyai tinggi badan sama

Source Code

datasetfutsal.java	
1.	<code>import java.util.*;</code>
2.	<code>class DataSetFutsal1 {</code>
3.	<code> int[] dataTinggiTimA = { 168, 170, 165, 168, 172,</code>
	<code>170, 169, 165, 171, 166 };</code>

```

4.      int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,
5.      66, 56, 72, 56 };
6.      int[] dataTinggiTimB = { 170, 167, 165, 166, 168,
7.      175, 172, 171, 168, 169 };
8.      int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,
9.      68, 68, 65, 60 };
10.
11.     public void BeratBadanSama() {
12.         Set<Integer> BeratBadanSama = new HashSet<>();
13.         int i = 0;
14.         do {
15.             int j = 0;
16.             do {
17.                 if (dataBeratBadanTimA[i] ==
18. dataBeratBadanTimB[j]) {
19.
20. BeratBadanSama.add(dataBeratBadanTimA[i]);
21.             }
22.             j++;
23.         } while (j < dataBeratBadanTimB.length);
24.         i++;
25.     } while (i < dataBeratBadanTimA.length);
26.     System.out.println("Pemain yang mempunyai berat
27. badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
28. "\n"
29.         + BeratBadanSama);
30. }
31. public void TinggiSama() {
32.     Set<Integer> tinggiSama = new HashSet<>();
33.     int i = 0;
34.     do {
35.         int j = 0;
36.         do {
37.             if (dataTinggiTimA[i] ==
38. dataTinggiTimB[j] &&
39. !tinggiSama.contains(dataTinggiTimA[i])) {
40.                 tinggiSama.add(dataTinggiTimA[i]);
41.             }
42.             j++;
43.         } while (j < dataTinggiTimB.length);
44.         i++;
45.     } while (i < dataTinggiTimA.length);
46.     System.out.println("Pemain yang mempunyai tinggi
47. badan yang sama ada " + tinggiSama.size() + ", yaitu:" +

```

	"\n"
38.	+ tinggiSama);
39.	}
40.	
41.	public void rentangtinggibadan() {
42.	int i = 0;
43.	int mintinggibadan = Integer.MAX_VALUE;
44.	do {
45.	if (dataTinggiTimA[i] < mintinggibadan) {
46.	mintinggibadan = dataTinggiTimA[i];
47.	}
48.	i++;
49.	} while (i < dataTinggiTimA.length);
50.	i = 0;
51.	do {
52.	if (dataTinggiTimB[i] < mintinggibadan) {
53.	mintinggibadan = dataTinggiTimB[i];
54.	}
55.	i++;
56.	} while (i < dataTinggiTimB.length);
57.	i = 0;
58.	int maxTinggi = Integer.MIN_VALUE;
59.	do {
60.	if (dataTinggiTimA[i] > maxTinggi) {
61.	maxTinggi = dataTinggiTimA[i];
62.	}
63.	i++;
64.	} while (i < dataTinggiTimA.length);
65.	i = 0;
66.	do {
67.	if (dataTinggiTimB[i] > maxTinggi) {
68.	maxTinggi = dataTinggiTimB[i];
69.	}
70.	i++;
71.	} while (i < dataTinggiTimB.length);
72.	System.out.println("Rentang nilai tinggi badan kedua tim adalah " + mintinggibadan + " hingga " + maxTinggi);
73.	}
74.	
75.	public void rentangberatbadan() {
76.	int minBeratBadan = Integer.MAX_VALUE;
77.	int i = 0;
78.	do {


```

79.         if (dataBeratBadanTimA[i] < minBeratBadan) {
80.             minBeratBadan = dataBeratBadanTimA[i];
81.         }
82.         i++;
83.     } while (i < dataBeratBadanTimA.length);
84.     i = 0;
85.     do {
86.         if (dataBeratBadanTimB[i] < minBeratBadan) {
87.             minBeratBadan = dataBeratBadanTimB[i];
88.         }
89.         i++;
90.     } while (i < dataBeratBadanTimB.length);
91.     int maxBerat = Integer.MIN_VALUE;
92.     i = 0;
93.     do {
94.         if (dataBeratBadanTimA[i] > maxBerat) {
95.             maxBerat = dataBeratBadanTimA[i];
96.         }
97.         i++;
98.     } while (i < dataBeratBadanTimA.length);
99.     i = 0;
100.    do {
101.        if (dataBeratBadanTimB[i] > maxBerat) {
102.            maxBerat = dataBeratBadanTimB[i];
103.        }
104.        i++;
105.    } while (i < dataBeratBadanTimB.length);
106.    System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
107.    }
108.
109.    public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
    {
110.        Set<Integer>    p    =    new HashSet<Integer>();
111.        int i = 0;
112.        do
113.        {
114.            boolean f = false;
115.            int j = 0;
116.            do {
117.                if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
118.                    f = true;

```

```
119.         break;
120.     }
121.         j++;
122.     }
123.         while (j < dataBeratBadanTimA.length);
124.             if (!f) {
125.                 p.add(dataBeratBadanTimB[i]);
126.             }
127.             i++;
128.         }
129.         while (i < dataBeratBadanTimB.length);
130.             System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
131.         }
132.
133.     public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
134.         Set<Integer>    p    =    new HashSet<Integer>();
135.         int i = 0;
136.
137.         do {
138.
139.             boolean f = false;
140.
141.             int j = 0;
142.
143.             do {
144.
145.                 if      (dataTinggiTimA[i]  == dataTinggiTimB[j])
{
146.                     f = true;
147.                     break;
148.                 }
149.
150.
151.                 j++;
152.
153.             }
154.             while (j < dataTinggiTimB.length);
155.                 if (!f) {
156.
157.                     p.add(dataTinggiTimA[i]);
158.                 }
159.                 i++;
```

```

160.     }
161.         while (i < dataTinggiTimA.length);
162.             System.out.println("Tinggi badan pada
tim A yang tidak ada pada tim B ada " + p);
163.     }
164. }
165.
166. public class datasetfutsal {
167.     public static void main(String[] args) {
168.         Scanner p = new Scanner(System.in);
169.         DataSetFutsal1 q = new DataSetFutsal1();
170.         q.TinggiSama();
171.         // q.BeratBadanSama();
172.         // q.rentangtinggibadan();
173.         // q.rentangberatbadan();
174.         //
q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
175.         //
q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
176.         p.close();
177.     }
178. }

```

Output

The screenshot shows a terminal window with the following command and output:

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> c:; cd 'c:\
.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessage'
a58c2d2cb\redhat.java\jdt_ws\praktikumkesepuluh_b549c8e3\bin' '
Pemain yang mempunyai tinggi badan yang sama ada 7, yaitu:
[165, 166, 168, 169, 170, 171, 172]
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>

```

Below the terminal is a text editor window titled "Dani Adr" showing the following text:

```

Dani Adrian
225150201111009

```

The text editor window also shows a status bar at the bottom with the following information: Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8.

Penjelasan

Method Tinggi Sama ();

Set<Integer> tinggiSama

Objek set, menyimpan tinggi badan yang sama secara unik

nested do-while loop

Membandingkan setiap elemen data tinggi tim A dengan setiap elemen data tinggi tim B

Setelah selesai mengecek seluruh elemen, kemudian mencetak jumlah pemain yang memiliki tinggi badan yang sama

Jika ditemukan tinggi badan yang sama serta belum ada di dalam tinggi sama. Tinggi tersebut ditambahkan ke dalam tinggi sama

Setelah selesai mengecek semua elemen, kemudian mencetak jumlah pemain yang memiliki tinggi badan yang sama.

Pertanyaan

- b. Yang mempunyai berat badan sama

Source Code

```
datasetfutsal.java
1.  import java.util.*;
2.
3.  class DataSetFutsal1 {
4.      int[] dataTinggiTimA = { 168, 170, 165, 168, 172, 170,
169, 165, 171, 166 };
5.      int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,
66, 56, 72, 56 };
6.      int[] dataTinggiTimB = { 170, 167, 165, 166, 168, 175,
172, 171, 168, 169 };
7.      int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,
68, 68, 65, 60 };
8.
9.      public void BeratBadanSama() {
10.          Set<Integer> BeratBadanSama = new HashSet<>();
11.          int i = 0;
12.          do {
13.              int j = 0;
14.              do {
```

```

15.         if (dataBeratBadanTimA[i] ==
dataBeratBadanTimB[j]) {
16.
BeratBadanSama.add(dataBeratBadanTimA[i]);
17.         }
18.         j++;
19.     } while (j < dataBeratBadanTimB.length);
20.     i++;
21. } while (i < dataBeratBadanTimA.length);
22.     System.out.println("Pemain yang mempunyai berat
badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
"\n"
23.         + BeratBadanSama);
24. }
25.
26.     public void TinggiSama() {
27.         Set<Integer> tinggiSama = new HashSet<>();
28.         int i = 0;
29.         do {
30.             int j = 0;
31.             do {
32.                 if (dataTinggiTimA[i] == dataTinggiTimB[j]
&& !tinggiSama.contains(dataTinggiTimA[i])) {
33.                     tinggiSama.add(dataTinggiTimA[i]);
34.                 }
35.                 j++;
36.             } while (j < dataTinggiTimB.length);
37.             i++;
38.         } while (i < dataTinggiTimA.length);
39.         System.out.println("Pemain yang mempunyai tinggi
badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
"\n"
40.             + tinggiSama);
41.     }
42.
43.     public void rentangtinggibadan() {
44.         int i = 0;
45.         int mintinggibadan = Integer.MAX_VALUE;
46.         do {
47.             if (dataTinggiTimA[i] < mintinggibadan) {
48.                 mintinggibadan = dataTinggiTimA[i];
49.             }
50.             i++;
51.         } while (i < dataTinggiTimA.length);

```

```
52.         i = 0;
53.         do {
54.             if (dataTinggiTimB[i] < mintinggibadan) {
55.                 mintinggibadan = dataTinggiTimB[i];
56.             }
57.             i++;
58.         } while (i < dataTinggiTimB.length);
59.         i = 0;
60.         int maxTinggi = Integer.MIN_VALUE;
61.         do {
62.             if (dataTinggiTimA[i] > maxTinggi) {
63.                 maxTinggi = dataTinggiTimA[i];
64.             }
65.             i++;
66.         } while (i < dataTinggiTimA.length);
67.         i = 0;
68.         do {
69.             if (dataTinggiTimB[i] > maxTinggi) {
70.                 maxTinggi = dataTinggiTimB[i];
71.             }
72.             i++;
73.         } while (i < dataTinggiTimB.length);
74.         System.out.println("Rentang nilai tinggi badan
kedua tim adalah " + mintinggibadan + " hingga " +
maxTinggi);
75.     }
76.
77.     public void rentangberatbadan() {
78.         int minBeratBadan = Integer.MAX_VALUE;
79.         int i = 0;
80.         do {
81.             if (dataBeratBadanTimA[i] < minBeratBadan) {
82.                 minBeratBadan = dataBeratBadanTimA[i];
83.             }
84.             i++;
85.         } while (i < dataBeratBadanTimA.length);
86.         i = 0;
87.         do {
88.             if (dataBeratBadanTimB[i] < minBeratBadan) {
89.                 minBeratBadan = dataBeratBadanTimB[i];
90.             }
91.             i++;
92.         } while (i < dataBeratBadanTimB.length);
93.         int maxBerat = Integer.MIN_VALUE;
```

```
94.         i = 0;
95.         do {
96.             if (dataBeratBadanTimA[i] > maxBerat) {
97.                 maxBerat = dataBeratBadanTimA[i];
98.             }
99.             i++;
100.        } while (i < dataBeratBadanTimA.length);
101.        i = 0;
102.        do {
103.            if (dataBeratBadanTimB[i] > maxBerat) {
104.                maxBerat = dataBeratBadanTimB[i];
105.            }
106.            i++;
107.        } while (i < dataBeratBadanTimB.length);
108.        System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
109.    }
110.
111.    public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA() {
112.        Set<Integer> p = new HashSet<Integer>();
113.        int i = 0;
114.        do
115.        {
116.            boolean f = false;
117.            int j = 0;
118.            do {
119.                if (dataBeratBadanTimB[i] ==
dataBeratBadanTimA[j]) {
120.                    f = true;
121.                    break;
122.                }
123.                j++;
124.            }
125.            while (j < dataBeratBadanTimA.length);
126.            if (!f) {
127.                p.add(dataBeratBadanTimB[i]);
128.            }
129.            i++;
130.        }
131.        while (i < dataBeratBadanTimB.length);
132.        System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133.    }
```

```
134
135 public void TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB()
136 {
137     Set<Integer>    p    =    new HashSet<Integer>();
138     int i = 0;
139     do {
140         boolean f = false;
141         int j = 0;
142         do {
143             if      (dataTinggiTimA[i] == dataTinggiTimB[j]) {
144                 f = true;
145                 break;
146             }
147             j++;
148         }
149         while (j < dataTinggiTimB.length);
150         if (!f) {
151             p.add(dataTinggiTimA[i]);
152         }
153         i++;
154     }
155     while (i < dataTinggiTimA.length);
156     System.out.println("Tinggi badan pada tim
157 A yang tidak ada pada tim B ada " + p);
158 }
159
160 public class datasetfutsal {
161     public static void main(String[] args) {
162         Scanner p = new Scanner(System.in);
163         DataSetFutsal1 q = new DataSetFutsal1();
164         // q.TinggiSama();
165         q.BeratBadanSama();
166         // q.rentangtinggibadan();
167         // q.rentangberatbadan();
168     }
169 }
```

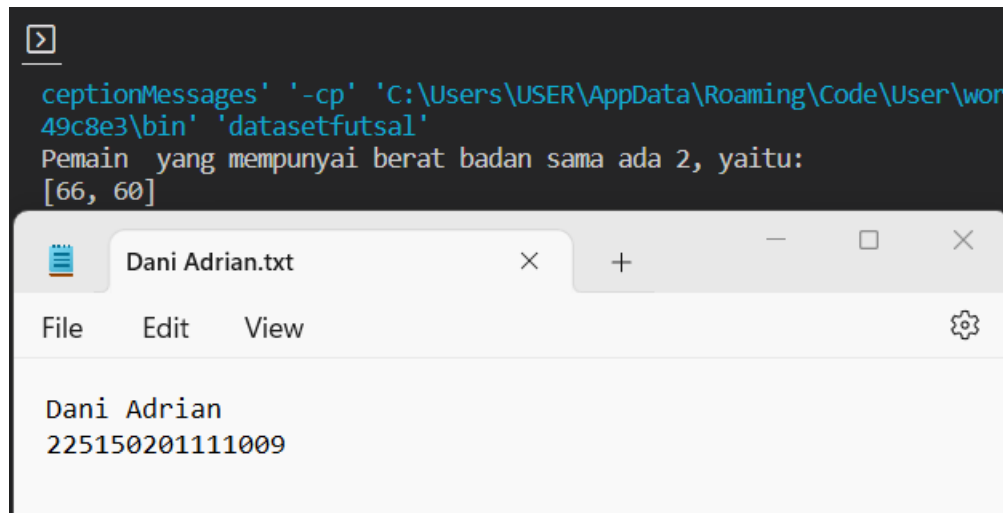


```

176      //
177      q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
178      // q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
179      p.close();
180  }

```

Output



The screenshot shows a code editor window with the following text:

```

ceptionMessages' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaces\datasetfutsal'
Pemain yang mempunyai berat badan sama ada 2, yaitu:
[66, 60]

```

Below the code editor is a text file named "Dani Adrian.txt" with the following content:

```

Dani Adrian
225150201111009

```

Penjelasan

Method Berat Badan Sama

nested do-while loop

Membandingkan setiap elemen dari data berat badan tim A dengan berat badan tim B

Jika ada pemain yang berat badan yang sama, maka pemain tersebut akan ditambahkan ke dalam berat badan sama, sebuah <Integer> untuk menghindari duplikat

Class dataSetFutsal

Objek datasetfutsal dengan nama q dibuat, kemudian method berat badan sama dari objek tersebut dipanggil

Pertanyaan

- c. Rentang nilai dari tinggi badan kedua tim

Source Code

datasetfutsal.java

```

1. import java.util.*;
2.
3. class DataSetFutsal1 {
4.     int[] dataTinggiTimA = { 168, 170, 165, 168, 172,
5.     170, 169, 165, 171, 166 };
6.     int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,
7.     66, 56, 72, 56 };
8.     int[] dataTinggiTimB = { 170, 167, 165, 166, 168,
9.     175, 172, 171, 168, 169 };
10.    int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,
11.    68, 68, 65, 60 };
12.
13.    public void BeratBadanSama() {
14.        Set<Integer> BeratBadanSama = new HashSet<>();
15.        int i = 0;
16.        do {
17.            int j = 0;
18.            do {
19.                if (dataBeratBadanTimA[i] ==
20.                dataBeratBadanTimB[j]) {
21.                    BeratBadanSama.add(dataBeratBadanTimA[i]);
22.                }
23.                j++;
24.            } while (j < dataBeratBadanTimB.length);
25.            i++;
26.        } while (i < dataBeratBadanTimA.length);
27.        System.out.println("Pemain yang mempunyai berat
28.        badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
29.        "\n"
30.        + BeratBadanSama);
31.    }
32.
33.    public void TinggiSama() {
34.        Set<Integer> tinggiSama = new HashSet<>();
35.        int i = 0;
36.        do {
37.            int j = 0;
38.            do {
39.                if (dataTinggiTimA[i] ==
40.                dataTinggiTimB[j] &&
41.                !tinggiSama.contains(dataTinggiTimA[i])) {

```

```

33.             tinggiSama.add(dataTinggiTimA[i]);
34.         }
35.         j++;
36.     } while (j < dataTinggiTimB.length);
37.     i++;
38. } while (i < dataTinggiTimA.length);
39. System.out.println("Pemain yang mempunyai tinggi
    badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
    "\n"
40.         + tinggiSama);
41. }
42.
43. public void rentangtinggibadan() {
44.     int i = 0;
45.     int mintinggibadan = Integer.MAX_VALUE;
46.     do {
47.         if (dataTinggiTimA[i] < mintinggibadan) {
48.             mintinggibadan = dataTinggiTimA[i];
49.         }
50.         i++;
51.     } while (i < dataTinggiTimA.length);
52.     i = 0;
53.     do {
54.         if (dataTinggiTimB[i] < mintinggibadan) {
55.             mintinggibadan = dataTinggiTimB[i];
56.         }
57.         i++;
58.     } while (i < dataTinggiTimB.length);
59.     i = 0;
60.     int maxTinggi = Integer.MIN_VALUE;
61.     do {
62.         if (dataTinggiTimA[i] > maxTinggi) {
63.             maxTinggi = dataTinggiTimA[i];
64.         }
65.         i++;
66.     } while (i < dataTinggiTimA.length);
67.     i = 0;
68.     do {
69.         if (dataTinggiTimB[i] > maxTinggi) {
70.             maxTinggi = dataTinggiTimB[i];
71.         }
72.         i++;
73.     } while (i < dataTinggiTimB.length);
74.     System.out.println("Rentang nilai tinggi badan

```

	kedua tim adalah " + mintinggibadan + " hingga " + maxTinggi);
75.	}
76.	
77.	public void rentangberatbadan() {
78.	int minBeratBadan = Integer.MAX_VALUE;
79.	int i = 0;
80.	do {
81.	if (dataBeratBadanTimA[i] < minBeratBadan) {
82.	minBeratBadan = dataBeratBadanTimA[i];
83.	}
84.	i++;
85.	} while (i < dataBeratBadanTimA.length);
86.	i = 0;
87.	do {
88.	if (dataBeratBadanTimB[i] < minBeratBadan) {
89.	minBeratBadan = dataBeratBadanTimB[i];
90.	}
91.	i++;
92.	} while (i < dataBeratBadanTimB.length);
93.	int maxBerat = Integer.MIN_VALUE;
94.	i = 0;
95.	do {
96.	if (dataBeratBadanTimA[i] > maxBerat) {
97.	maxBerat = dataBeratBadanTimA[i];
98.	}
99.	i++;
100.	} while (i < dataBeratBadanTimA.length);
101.	i = 0;
102.	do {
103.	if (dataBeratBadanTimB[i] > maxBerat) {
104.	maxBerat = dataBeratBadanTimB[i];
105.	}
106.	i++;
107.	} while (i < dataBeratBadanTimB.length);
108.	System.out.println("Rentang nilai berat badan kedua tim adalah " + minBeratBadan + " hingga " + maxBerat);
109.	}
110.	
111.	public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
	{
112.	Set<Integer> p = new HashSet<Integer>();
113.	int i = 0;

```
114.         do
115.         {
116.             boolean f = false;
117.             int j = 0;
118.             do {
119.                 if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
120.                     f = true;
121.                     break;
122.                 }
123.                 j++;
124.             }
125.             while (j < dataBeratBadanTimA.length);
126.             if (!f) {
127.                 p.add(dataBeratBadanTimB[i]);
128.             }
129.             i++;
130.         }
131.         while (i < dataBeratBadanTimB.length);
132.         System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133.     }
134.
135.     public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
136.         Set<Integer>    p      =    new HashSet<Integer>();
137.         int i = 0;
138.
139.         do {
140.
141.             boolean f = false;
142.
143.             int j = 0;
144.
145.             do {
146.
147.                 if      (dataTinggiTimA[i]    == dataTinggiTimB[j])
{
148.                     f = true;
149.                     break;
150.                 }
151.             }
152.
153.             j++;
```

```
154.
155.     }
156.     while (j < dataTinggiTimB.length);
157.         if (!f) {
158.
159.             p.add(dataTinggiTimA[i]);
160.         }
161.         i++;
162.     }
163.         while (i < dataTinggiTimA.length);
164.             System.out.println("Tinggi badan pada tim
A yang tidak ada pada tim B ada " + p);
165.         }
166.     }
167.
168. public class datasetfutsal {
169.     public static void main(String[] args) {
170.         Scanner p = new Scanner(System.in);
171.         DataSetFutsal1 q = new DataSetFutsal1();
172.         // q.TinggiSama();
173.         // q.BeratBadanSama();
174.         q.rentangtinggibadan();
175.         // q.rentangberatbadan();
176.         //
177.         q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
178.         //
179.         q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
180.         p.close();
181.     }
182. }
```

Output

```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> c:; cd 'c:\U\n\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMess'
Rentang nilai tinggi badan kedua tim adalah 165 hingga 175
```

Dani Adrian.txt

Dani Adrian
225150201111009

Penjelasan

Method rentang tinggi badan

Terdapat beberapa do-while loop : untuk mencari nilai minimum dan maksimum dari tinggi badan kedua tim

Rentang nilai tinggi badan kemudian dicetak dengan `System.out.println()`

Class datasetfutsal

Method `main()`, objek dari class `datasetfutsal` dengan nama `q` dibuat, dan kemudian method rentang tinggi badan dipanggil melalui objek tersebut

Pertanyaan

- d. Rentang nilai dari berat badan kedua tim

Source Code

datasetfutsal.java	
1.	<code>import java.util.*;</code>
2.	
3.	<code>class DataSetFutsal1 {</code>
4.	<code>int[] dataTinggiTimA = { 168, 170, 165, 168, 172,</code>
	<code>170, 169, 165, 171, 166 };</code>
5.	<code>int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,</code>
	<code>66, 56, 72, 56 };</code>
6.	<code>int[] dataTinggiTimB = { 170, 167, 165, 166, 168,</code>
	<code>175, 172, 171, 168, 169 };</code>
7.	<code>int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,</code>
	<code>68, 68, 65, 60 };</code>
8.	
9.	<code>public void BeratBadanSama() {</code>

```

10.         Set<Integer> BeratBadanSama = new HashSet<>();
11.         int i = 0;
12.         do {
13.             int j = 0;
14.             do {
15.                 if (dataBeratBadanTimA[i] ==
dataBeratBadanTimB[j]) {
16.                     BeratBadanSama.add(dataBeratBadanTimA[i]);
17.                 }
18.                 j++;
19.             } while (j < dataBeratBadanTimB.length);
20.             i++;
21.         } while (i < dataBeratBadanTimA.length);
22.         System.out.println("Pemain yang mempunyai berat
badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
"\n"
23.             + BeratBadanSama);
24.     }
25.
26.     public void TinggiSama() {
27.         Set<Integer> tinggiSama = new HashSet<>();
28.         int i = 0;
29.         do {
30.             int j = 0;
31.             do {
32.                 if (dataTinggiTimA[i] ==
dataTinggiTimB[j] &&
!tinggiSama.contains(dataTinggiTimA[i])) {
33.                     tinggiSama.add(dataTinggiTimA[i]);
34.                 }
35.                 j++;
36.             } while (j < dataTinggiTimB.length);
37.             i++;
38.         } while (i < dataTinggiTimA.length);
39.         System.out.println("Pemain yang mempunyai tinggi
badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
"\n"
40.             + tinggiSama);
41.     }
42.
43.     public void rentangtinggibadan() {
44.         int i = 0;
45.         int mintinggibadan = Integer.MAX_VALUE;

```



```

46.         do {
47.             if (dataTinggiTimA[i] < mintinggibadan) {
48.                 mintinggibadan = dataTinggiTimA[i];
49.             }
50.             i++;
51.         } while (i < dataTinggiTimA.length);
52.         i = 0;
53.         do {
54.             if (dataTinggiTimB[i] < mintinggibadan) {
55.                 mintinggibadan = dataTinggiTimB[i];
56.             }
57.             i++;
58.         } while (i < dataTinggiTimB.length);
59.         i = 0;
60.         int maxTinggi = Integer.MIN_VALUE;
61.         do {
62.             if (dataTinggiTimA[i] > maxTinggi) {
63.                 maxTinggi = dataTinggiTimA[i];
64.             }
65.             i++;
66.         } while (i < dataTinggiTimA.length);
67.         i = 0;
68.         do {
69.             if (dataTinggiTimB[i] > maxTinggi) {
70.                 maxTinggi = dataTinggiTimB[i];
71.             }
72.             i++;
73.         } while (i < dataTinggiTimB.length);
74.         System.out.println("Rentang nilai tinggi badan
kedua tim adalah " + mintinggibadan + " hingga " +
maxTinggi);
75.     }
76.
77.     public void rentangberatbadan() {
78.         int minBeratBadan = Integer.MAX_VALUE;
79.         int i = 0;
80.         do {
81.             if (dataBeratBadanTimA[i] < minBeratBadan) {
82.                 minBeratBadan = dataBeratBadanTimA[i];
83.             }
84.             i++;
85.         } while (i < dataBeratBadanTimA.length);
86.         i = 0;
87.         do {

```

```

88.         if (dataBeratBadanTimB[i] < minBeratBadan) {
89.             minBeratBadan = dataBeratBadanTimB[i];
90.         }
91.         i++;
92.     } while (i < dataBeratBadanTimB.length);
93.     int maxBerat = Integer.MIN_VALUE;
94.     i = 0;
95.     do {
96.         if (dataBeratBadanTimA[i] > maxBerat) {
97.             maxBerat = dataBeratBadanTimA[i];
98.         }
99.         i++;
100.    } while (i < dataBeratBadanTimA.length);
101.    i = 0;
102.    do {
103.        if (dataBeratBadanTimB[i] > maxBerat) {
104.            maxBerat = dataBeratBadanTimB[i];
105.        }
106.        i++;
107.    } while (i < dataBeratBadanTimB.length);
108.    System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
109.    }
110.
111. public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
    {
112.     Set<Integer>    p    =    new HashSet<Integer>();
113.     int i = 0;
114.     do
115.     {
116.         boolean f = false;
117.         int j = 0;
118.         do {
119.             if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
120.                 f = true;
121.                 break;
122.             }
123.             j++;
124.         }
125.         while (j < dataBeratBadanTimA.length);
126.         if (!f) {
127.             p.add(dataBeratBadanTimB[i]);

```

```
128. }
129.     i++;
130. }
131.     while (i < dataBeratBadanTimB.length);
132.         System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133.     }
134.
135. public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
136.     Set<Integer>    p    =    new HashSet<Integer>();
137.     int i = 0;
138.
139.         do {
140.
141.             boolean f = false;
142.
143.             int j = 0;
144.
145.                 do {
146.
147.                     if      (dataTinggiTimA[i]  == dataTinggiTimB[j])
{
148.                         f = true;
149.                         break;
150.
151.                     }
152.
153.                     j++;
154.
155.                 }
156.                 while (j < dataTinggiTimB.length);
157.                 if (!f) {
158.
159.                     p.add(dataTinggiTimA[i]);
160.                 }
161.                 i++;
162.             }
163.             while (i < dataTinggiTimA.length);
164.             System.out.println("Tinggi badan pada tim
A yang tidak ada pada tim B ada " + p);
165.         }
166.     }
167.
```

```

168. public class datasetfutsal {
169.     public static void main(String[] args) {
170.         Scanner p = new Scanner(System.in);
171.         DataSetFutsal1 q = new DataSetFutsal1();
172.         // q.TinggiSama();
173.         // q.BeratBadanSama();
174.         // q.rentangtinggibadan();
175.         q.rentangberatbadan();
176.         //
177.         q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
178.         //
179.         q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
180.         p.close();
181.     }
182. }

```

Output

```

s\Java\jdk-20\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetails'
er\workspaceStorage\02422310aed8e2164ebf10ca58c2d2cb\redhat.java\
Rentang nilai berat badan kedua tim adalah 50 hingga 72
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>

```

Dani Adrian.txt

File Edit View

Dani Adrian
225150201111009

Penjelasan

Method rentang berat badan

Variabel minimum berat badan diinisialisasi dengan `Integer.MAX_VALUE` (nilai terbesar yang diwakili oleh tipe data `int`) untuk mencari nilai berat badan terkecil.

do-while loop, setiap elemen dalam array data berat badan tim A dan data berat badan tim B diperiksa. Jika nilai berat badan, saat ini lebih kecil dari `minBerat`, maka nilai `minBerat` diperbarui.

Variabel `maxBerat` diinisialisasi dengan `Integer.MIN_VALUE` (nilai terkecil yang dapat diwakili oleh tipe data `int`) untuk mencari nilai berat badan terbesar.

do-while loop, setiap elemen dalam array data berat badan tim A dan data berat badan tim B diperiksa. Jika nilai berat badan saat ini lebih besar dari maxBerat, maka nilai maxBerat diperbarui.

Setelah mengintegrasikan kedua array, rentang nilai berat badan dari kedua tim dicetak ke layar menggunakan `System.out.println()`.

Class datasetfutsal

Pada method `main()`, objek dari kelas data set futsal dibuat dan metode rentang berat badan dipanggil untuk menghitung dan mencetak rentang nilai berat badan kedua tim

Pertanyaan

- e. Tinggi badan pada tim A yang tidak ada pada tim B

Source Code

datasetfutsal.java	
1.	<code>import java.util.*;</code>
2.	
3.	<code>class DataSetFutsal1 {</code>
4.	<code> int[] dataTinggiTimA = { 168, 170, 165, 168, 172,</code>
	<code>170, 169, 165, 171, 166 };</code>
5.	<code> int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,</code>
	<code>66, 56, 72, 56 };</code>
6.	<code> int[] dataTinggiTimB = { 170, 167, 165, 166, 168,</code>
	<code>175, 172, 171, 168, 169 };</code>
7.	<code> int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,</code>
	<code>68, 68, 65, 60 };</code>
8.	
9.	<code> public void BeratBadanSama() {</code>
10.	<code> Set<Integer> BeratBadanSama = new HashSet<>();</code>
11.	<code> int i = 0;</code>
12.	<code> do {</code>
13.	<code> int j = 0;</code>
14.	<code> do {</code>
15.	<code> if (dataBeratBadanTimA[i] ==</code>
	<code>dataBeratBadanTimB[j]) {</code>
16.	<code>BeratBadanSama.add(dataBeratBadanTimA[i]);</code>
17.	<code> }</code>

```

18.         j++;
19.         } while (j < dataBeratBadanTimB.length);
20.         i++;
21.     } while (i < dataBeratBadanTimA.length);
22.     System.out.println("Pemain yang mempunyai berat
    badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
    "\n"
23.         + BeratBadanSama);
24. }
25.
26.     public void TinggiSama() {
27.         Set<Integer> tinggiSama = new HashSet<>();
28.         int i = 0;
29.         do {
30.             int j = 0;
31.             do {
32.                 if (dataTinggiTimA[i] ==
    dataTinggiTimB[j] &&
    !tinggiSama.contains(dataTinggiTimA[i])) {
33.                     tinggiSama.add(dataTinggiTimA[i]);
34.                 }
35.                 j++;
36.             } while (j < dataTinggiTimB.length);
37.             i++;
38.         } while (i < dataTinggiTimA.length);
39.         System.out.println("Pemain yang mempunyai tinggi
    badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
    "\n"
40.             + tinggiSama);
41.     }
42.
43.     public void rentangtinggibadan() {
44.         int i = 0;
45.         int mintinggibadan = Integer.MAX_VALUE;
46.         do {
47.             if (dataTinggiTimA[i] < mintinggibadan) {
48.                 mintinggibadan = dataTinggiTimA[i];
49.             }
50.             i++;
51.         } while (i < dataTinggiTimA.length);
52.         i = 0;
53.         do {
54.             if (dataTinggiTimB[i] < mintinggibadan) {
55.                 mintinggibadan = dataTinggiTimB[i];

```

```

56.         }
57.         i++;
58.     } while (i < dataTinggiTimB.length);
59.     i = 0;
60.     int maxTinggi = Integer.MIN_VALUE;
61.     do {
62.         if (dataTinggiTimA[i] > maxTinggi) {
63.             maxTinggi = dataTinggiTimA[i];
64.         }
65.         i++;
66.     } while (i < dataTinggiTimA.length);
67.     i = 0;
68.     do {
69.         if (dataTinggiTimB[i] > maxTinggi) {
70.             maxTinggi = dataTinggiTimB[i];
71.         }
72.         i++;
73.     } while (i < dataTinggiTimB.length);
74.     System.out.println("Rentang nilai tinggi badan
kedua tim adalah " + mintinggibadan + " hingga " +
maxTinggi);
75.     }
76.
77.     public void rentangberatbadan() {
78.         int minBeratBadan = Integer.MAX_VALUE;
79.         int i = 0;
80.         do {
81.             if (dataBeratBadanTimA[i] < minBeratBadan) {
82.                 minBeratBadan = dataBeratBadanTimA[i];
83.             }
84.             i++;
85.         } while (i < dataBeratBadanTimA.length);
86.         i = 0;
87.         do {
88.             if (dataBeratBadanTimB[i] < minBeratBadan) {
89.                 minBeratBadan = dataBeratBadanTimB[i];
90.             }
91.             i++;
92.         } while (i < dataBeratBadanTimB.length);
93.         int maxBerat = Integer.MIN_VALUE;
94.         i = 0;
95.         do {
96.             if (dataBeratBadanTimA[i] > maxBerat) {
97.                 maxBerat = dataBeratBadanTimA[i];

```

```

98.         }
99.         i++;
100.    } while (i < dataBeratBadanTimA.length);
101.    i = 0;
102.    do {
103.        if (dataBeratBadanTimB[i] > maxBerat) {
104.            maxBerat = dataBeratBadanTimB[i];
105.        }
106.        i++;
107.    } while (i < dataBeratBadanTimB.length);
108.    System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
109.    }
110.
111.    public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
    {
112.        Set<Integer>    p    =    new HashSet<Integer>();
113.        int i = 0;
114.        do
115.        {
116.            boolean f = false;
117.            int j = 0;
118.            do {
119.                if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
120.                    f = true;
121.                    break;
122.                }
123.                j++;
124.            }
125.            while (j < dataBeratBadanTimA.length);
126.            if (!f) {
127.                p.add(dataBeratBadanTimB[i]);
128.            }
129.            i++;
130.        }
131.        while (i < dataBeratBadanTimB.length);
132.        System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133.    }
134.
135.    public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {

```



```

136.      Set<Integer>    p      =    new HashSet<Integer>();
137.      int i = 0;
138.
139.          do {
140.
141.              boolean f = false;
142.
143.              int j = 0;
144.
145.                  do {
146.
147.                      if      (dataTinggiTimA[i]  == dataTinggiTimB[j])
148.                  {
149.                      f = true;
150.                      break;
151.                  }
152.
153.                      j++;
154.
155.                  }
156.              while (j  < dataTinggiTimB.length);
157.                  if (!f) {
158.
159.                      p.add(dataTinggiTimA[i]);
160.                  }
161.                      i++;
162.                  }
163.                  while (i <  dataTinggiTimA.length);
164.                      System.out.println("Tinggi badan pada
tim A yang tidak ada pada tim B ada " + p);
165.                  }
166.              }
167.
168.      public class datasetfutsal {
169.          public static void main(String[] args) {
170.              Scanner p = new Scanner(System.in);
171.              DataSetFutsal1 q = new DataSetFutsal1();
172.              // q.TinggiSama();
173.              // q.BeratBadanSama();
174.              // q.rentangtinggibadan();
175.              // q.rentangberatbadan();
176.
177.              q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();

```

177.	//
178.	q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
179.	p.close();
180.	}

Output

The screenshot shows a Java IDE with a command prompt window and a text file named 'Dani Adrian.txt'.

Command Prompt Output:

```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> c:: cd 'c:\Users\USER\JAVA V
a.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users
ca58c2d2cb\redhat.java\jdt_ws\praktikumkesepuluh_b549c8e3\bin' 'datasetfutsal'
Tinggi badan pada tim A yang tidak ada pada tim B ada []
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>
```

Text File Content (Dani Adrian.txt):

```
Dani Adrian
225150201111009
```

Penjelasan

Method Tinggi Badan tim A tidak ada pada Tinggi Badan tim B

Loop do-while untuk memeriksa setiap elemen dalam array data Tinggi TimA dan membandingkannya dengan setiap elemen dalam array dataTinggiTimB.

Jika ditemukan elemen yang sama, variabel boolean f akan diubah menjadi true. Jika tidak ada elemen yang sama, maka elemen tersebut ditambahkan ke dalam objek Set p.

Class datasetfutsal

Objek datasetfutsal dengan nama q dibuat, dan kemudian metode Tinggi Badan tim A yang tidak ada pada_tinggi Badan tim B dari objek tersebut dipanggil untuk menjalankannya.

Pertanyaan

- Berat badan pada tim B yang tidak ada pada tim A

Source Code

datasetfutsal.java

```

1.  import java.util.*;
2.
3.  class DataSetFutsal1 {
4.      int[] dataTinggiTimA = { 168, 170, 165, 168, 172,
5.      170, 169, 165, 171, 166 };
6.      int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,
7.      66, 56, 72, 56 };
8.      int[] dataTinggiTimB = { 170, 167, 165, 166, 168,
9.      175, 172, 171, 168, 169 };
10.     int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,
11.     68, 68, 65, 60 };
12.
13.     public void BeratBadanSama() {
14.         Set<Integer> BeratBadanSama = new HashSet<>();
15.         int i = 0;
16.         do {
17.             int j = 0;
18.             do {
19.                 if (dataBeratBadanTimA[i] ==
20.                 dataBeratBadanTimB[j]) {
21.                     BeratBadanSama.add(dataBeratBadanTimA[i]);
22.                     j++;
23.                 } while (j < dataBeratBadanTimB.length);
24.                 i++;
25.             } while (i < dataBeratBadanTimA.length);
26.             System.out.println("Pemain yang mempunyai berat
27.             badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
28.             "\n"
29.             + BeratBadanSama);
30.         }
31.
32.         public void TinggiSama() {
33.             Set<Integer> tinggiSama = new HashSet<>();
34.             int i = 0;
35.             do {
36.                 int j = 0;
37.                 do {
38.                     if (dataTinggiTimA[i] ==
39.                     dataTinggiTimB[j] &&
40.                     !tinggiSama.contains(dataTinggiTimA[i])) {

```

33.	tinggiSama.add(dataTinggiTimA[i]);
34.	}
35.	j++;
36.	} while (j < dataTinggiTimB.length);
37.	i++;
38.	} while (i < dataTinggiTimA.length);
39.	System.out.println("Pemain yang mempunyai tinggi badan yang sama ada " + tinggiSama.size() + ", yaitu:" + "\n"
40.	+ tinggiSama);
41.	}
42.	
43.	public void rentangtinggibadan() {
44.	int i = 0;
45.	int mintinggibadan = Integer.MAX_VALUE;
46.	do {
47.	if (dataTinggiTimA[i] < mintinggibadan) {
48.	mintinggibadan = dataTinggiTimA[i];
49.	}
50.	i++;
51.	} while (i < dataTinggiTimA.length);
52.	i = 0;
53.	do {
54.	if (dataTinggiTimB[i] < mintinggibadan) {
55.	mintinggibadan = dataTinggiTimB[i];
56.	}
57.	i++;
58.	} while (i < dataTinggiTimB.length);
59.	i = 0;
60.	int maxTinggi = Integer.MIN_VALUE;
61.	do {
62.	if (dataTinggiTimA[i] > maxTinggi) {
63.	maxTinggi = dataTinggiTimA[i];
64.	}
65.	i++;
66.	} while (i < dataTinggiTimA.length);
67.	i = 0;
68.	do {
69.	if (dataTinggiTimB[i] > maxTinggi) {
70.	maxTinggi = dataTinggiTimB[i];
71.	}
72.	i++;
73.	} while (i < dataTinggiTimB.length);
74.	System.out.println("Rentang nilai tinggi badan

	kedua tim adalah " + mintinggibadan + " hingga " + maxTinggi);
75.	}
76.	
77.	public void rentangberatbadan() {
78.	int minBeratBadan = Integer.MAX_VALUE;
79.	int i = 0;
80.	do {
81.	if (dataBeratBadanTimA[i] < minBeratBadan) {
82.	minBeratBadan = dataBeratBadanTimA[i];
83.	}
84.	i++;
85.	} while (i < dataBeratBadanTimA.length);
86.	i = 0;
87.	do {
88.	if (dataBeratBadanTimB[i] < minBeratBadan) {
89.	minBeratBadan = dataBeratBadanTimB[i];
90.	}
91.	i++;
92.	} while (i < dataBeratBadanTimB.length);
93.	int maxBerat = Integer.MIN_VALUE;
94.	i = 0;
95.	do {
96.	if (dataBeratBadanTimA[i] > maxBerat) {
97.	maxBerat = dataBeratBadanTimA[i];
98.	}
99.	i++;
100.	} while (i < dataBeratBadanTimA.length);
101.	i = 0;
102.	do {
103.	if (dataBeratBadanTimB[i] > maxBerat) {
104.	maxBerat = dataBeratBadanTimB[i];
105.	}
106.	i++;
107.	} while (i < dataBeratBadanTimB.length);
108.	System.out.println("Rentang nilai berat badan kedua tim adalah " + minBeratBadan + " hingga " + maxBerat);
109.	}
110.	
111.	public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
	{
112.	Set<Integer> p = new HashSet<Integer>();
113.	int i = 0;

```
114.         do
115.         {
116.             boolean f = false;
117.             int j = 0;
118.             do {
119.                 if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
120.                     f = true;
121.                     break;
122.                 }
123.                 j++;
124.             }
125.             while (j < dataBeratBadanTimA.length);
126.             if (!f) {
127.                 p.add(dataBeratBadanTimB[i]);
128.             }
129.             i++;
130.         }
131.         while (i < dataBeratBadanTimB.length);
132.         System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133.     }
134.
135.     public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
136.         Set<Integer>    p    =    new HashSet<Integer>();
137.         int i = 0;
138.
139.         do {
140.
141.             boolean f = false;
142.
143.             int j = 0;
144.
145.             do {
146.
147.                 if      (dataTinggiTimA[i]    == dataTinggiTimB[j])
{
148.                     f = true;
149.                     break;
150.                 }
151.             }
152.
153.             j++;
```

```

154.
155.     }
156.     while (j < dataTinggiTimB.length);
157.         if (!f) {
158.
159.             p.add(dataTinggiTimA[i]);
160.         }
161.         i++;
162.     }
163.         while (i < dataTinggiTimA.length);
164.             System.out.println("Tinggi badan pada
tim A yang tidak ada pada tim B ada " + p);
165.         }
166.     }
167.
168. public class datasetfutsal {
169.     public static void main(String[] args) {
170.         Scanner p = new Scanner(System.in);
171.         DataSetFutsal1 q = new DataSetFutsal1();
172.         // q.TinggiSama();
173.         // q.BeratBadanSama();
174.         // q.rentangtinggibadan();
175.         // q.rentangberatbadan();
176.         //
q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
177.         q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
178.         p.close();
179.     }
180. }

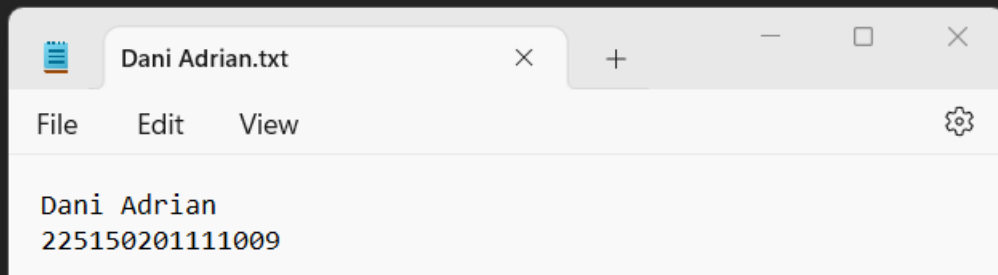
```

Output

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> c:: cd 'c:\Users\USER\JAVA
\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
10aed8e2164ebf10ca58c2d2cb\redhat.java\jdt_ws\praktikumkesepuluh_b549c8e3\bin' 'c
Berat badan pada tim B yang tidak ada pada tim A ada [65, 68, 71, 58, 59]
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>

```



Penjelasan

Method Berat badan tim B tidak ada pada berat badan tim A

Method ini menggunakan logika looping dengan do-while untuk membandingkan setiap elemen dalam array dataBerat Badan Tim B dengan elemen-elemen dalam array data Berat Badan Tim A.

Jika terdapat elemen dalam data berat badan Tim B yang tidak ada dalam data Berat Badan Tim A, maka elemen tersebut akan ditambahkan ke dalam Set dengan nama p.

Class datasetfutsal

Objek dari class datasetfutsal dengan nama q dibuat, kemudian method berat badan Tim B tidak ada pada berat badan tim A dipanggil pada objek tersebut.

Pertanyaan

2. Buatlah implementasi Map dalam program java berdasarkan kondisi berikut :
 - a. Implementasikan Map untuk menyimpan data tim A dan tim B dalam bentuk map terpisah.

Source code

datasetfutsal.java	
1.	import java.util.*;
2.	
3.	class DataSetFutsal1 {
4.	int[] dataTinggiTimA = { 168, 170, 165, 168, 172, 170, 169, 165, 171, 166 };
5.	int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70, 66, 56, 72, 56 };
6.	int[] dataTinggiTimB = { 170, 167, 165, 166, 168, 175, 172, 171, 168, 169 };
7.	int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71, 68, 68, 65, 60 };
8.	
9.	public void BeratBadanSama() {
10.	Set<Integer> BeratBadanSama = new HashSet<>();
11.	int i = 0;
12.	do {


```

13.         int j = 0;
14.         do {
15.             if (dataBeratBadanTimA[i] ==
dataBeratBadanTimB[j]) {
16. BeratBadanSama.add(dataBeratBadanTimA[i]);
17.             }
18.             j++;
19.         } while (j < dataBeratBadanTimB.length);
20.         i++;
21.     } while (i < dataBeratBadanTimA.length);
22.     System.out.println("Pemain yang mempunyai berat
badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
"\n"
23.         + BeratBadanSama);
24. }

25.
26.     public void TinggiSama() {
27.         Set<Integer> tinggiSama = new HashSet<>();
28.         int i = 0;
29.         do {
30.             int j = 0;
31.             do {
32.                 if (dataTinggiTimA[i] ==
dataTinggiTimB[j] &&
!tinggiSama.contains(dataTinggiTimA[i])) {
33.                     tinggiSama.add(dataTinggiTimA[i]);
34.                 }
35.                 j++;
36.             } while (j < dataTinggiTimB.length);
37.             i++;
38.         } while (i < dataTinggiTimA.length);
39.         System.out.println("Pemain yang mempunyai tinggi
badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
"\n"
40.             + tinggiSama);
41.     }

42.
43.     public void rentangtinggibadan() {
44.         int i = 0;
45.         int mintinggibadan = Integer.MAX_VALUE;
46.         do {
47.             if (dataTinggiTimA[i] < mintinggibadan) {
48.                 mintinggibadan = dataTinggiTimA[i];

```

```

49.         }
50.         i++;
51.     } while (i < dataTinggiTimA.length);
52.     i = 0;
53.     do {
54.         if (dataTinggiTimB[i] < mintinggibadan) {
55.             mintinggibadan = dataTinggiTimB[i];
56.         }
57.         i++;
58.     } while (i < dataTinggiTimB.length);
59.     i = 0;
60.     int maxTinggi = Integer.MIN_VALUE;
61.     do {
62.         if (dataTinggiTimA[i] > maxTinggi) {
63.             maxTinggi = dataTinggiTimA[i];
64.         }
65.         i++;
66.     } while (i < dataTinggiTimA.length);
67.     i = 0;
68.     do {
69.         if (dataTinggiTimB[i] > maxTinggi) {
70.             maxTinggi = dataTinggiTimB[i];
71.         }
72.         i++;
73.     } while (i < dataTinggiTimB.length);
74.     System.out.println("Rentang nilai tinggi badan
kedua tim adalah " + mintinggibadan + " hingga " +
maxTinggi);
75.     }
76.
77.     public void rentangberatbadan() {
78.         int minBeratBadan = Integer.MAX_VALUE;
79.         int i = 0;
80.         do {
81.             if (dataBeratBadanTimA[i] < minBeratBadan) {
82.                 minBeratBadan = dataBeratBadanTimA[i];
83.             }
84.             i++;
85.         } while (i < dataBeratBadanTimA.length);
86.         i = 0;
87.         do {
88.             if (dataBeratBadanTimB[i] < minBeratBadan) {
89.                 minBeratBadan = dataBeratBadanTimB[i];
90.             }

```

```

91.         i++;
92.     } while (i < dataBeratBadanTimB.length);
93.     int maxBerat = Integer.MIN_VALUE;
94.     i = 0;
95.     do {
96.         if (dataBeratBadanTimA[i] > maxBerat) {
97.             maxBerat = dataBeratBadanTimA[i];
98.         }
99.         i++;
100.    } while (i < dataBeratBadanTimA.length);
101.    i = 0;
102.    do {
103.        if (dataBeratBadanTimB[i] > maxBerat) {
104.            maxBerat = dataBeratBadanTimB[i];
105.        }
106.        i++;
107.    } while (i < dataBeratBadanTimB.length);
108.    System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
109.    }
110.
111. public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
    {
112.     Set<Integer>    p    =    new HashSet<Integer>();
113.     int i = 0;
114.     do
115.     {
116.         boolean f = false;
117.         int j = 0;
118.         do {
119.             if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
120.                 f = true;
121.                 break;
122.             }
123.             j++;
124.         }
125.         while (j < dataBeratBadanTimA.length);
126.         if (!f) {
127.             p.add(dataBeratBadanTimB[i]);
128.         }
129.         i++;
130.     }

```

```

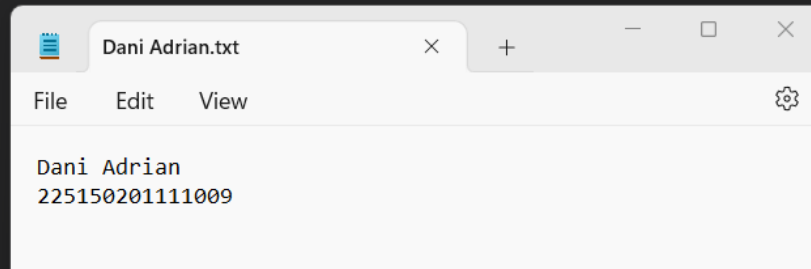
131.         while (i < dataBeratBadanTimB.length);
132.             System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133.         }
134.
135.     public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
136.         Set<Integer>    p    =    new HashSet<Integer>();
137.         int i = 0;
138.         do {
139.             boolean f = false;
140.             int j = 0;
141.             do {
142.                 if      (dataTinggiTimA[i]  == dataTinggiTimB[j])
{
143.                     f = true;
144.                     break;
145.                 }
146.             }
147.
148.             j++;
149.
150.         }
151.         while (j < dataTinggiTimB.length);
152.         if (!f) {
153.
154.             p.add(dataTinggiTimA[i]);
155.         }
156.         i++;
157.     }
158.         while (i < dataTinggiTimA.length);
159.             System.out.println("Tinggi badan pada tim
A yang tidak ada pada tim B ada " + p);
160.         }
161.
162.     public void SimpanDatauntukTimAdanTimB() {
163.         Map<Integer, Integer> timA = new HashMap<>();
164.         int i = 0;
165.         do {
166.             timA.put(dataTinggiTimA[i],
dataBeratBadanTimA[i]);
167.             i++;
168.
169.         } while (i < dataTinggiTimA.length);

```

```
170.
171. Map<Integer, Integer> timB = new HashMap<>();
172.     int j = 0;
173.     do
174.     {
175.         timB.put(dataTinggiTimB[j],
176. dataBeratBadanTimB[j]);
177.         j++;
178.     }
179.     while (j < dataTinggiTimB.length);
180.     System.out.println("Data tinggi dan berat
    badan Tim A: " + timA);
181.     System.out.println("Data tinggi berat badan
    Tim B: " + timB);
182. }
183. public class datasetfutsal {
184.     public static void main(String[] args) {
185.         Scanner p = new Scanner(System.in);
186.         DataSetFutsal1 q = new DataSetFutsal1();
187.         // q.TinggiSama();
188.         // q.BeratBadanSama();
189.         // q.rentangtinggibadan();
190.         // q.rentangberatbadan();
191.
192.         //q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
193.         //
194.         q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
195.         q.SimpanDatauntukTimAdanTimB();
196.         p.close();
197.     }
198. }
```

Output

```
PS C:\Users\USER\JAVA VS COD> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh'; & 'C:\Program
' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStora
dt_ws\praktikumkesepuluh_b549c8e3\bin' 'datasetfutsal'
Data tinggi dan berat badan Tim A: {165=56, 166=56, 168=55, 169=66, 170=70, 171=72, 172=60}
Data tinggi berat badan Tim B: {165=59, 166=58, 167=60, 168=65, 169=60, 170=66, 171=68, 172=68, 175=71}
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> █
```



Penjelasan

Method simpan Data TimA TimB

- Dibuat dua objek Map yaitu timA dan timB yang berturut-turut akan menyimpan data tinggi sebagai kunci dan berat badan sebagai nilai.
- Method ini menggunakan do-while loop untuk mengiterasi melalui array data dan menambahkan pasangan kunci-nilai ke dalam Map sesuai dengan indeks yang sedang diiterasi.

Class datasetfutsal

- Objek Data_SetFutsal dibuat dengan menggunakan operator new. Kemudian, metode simpanData TimA TimB dipanggil pada objek tersebut untuk menyimpan data tim.
- Pada saat program dijalankan, method simpanData_TimA_TimB() akan mencetak data tinggi dan berat badan dari Tim A dan Tim B menggunakan System.out.println(). Data dari Tim A dicetak dengan menggunakan objek Map timA, sedangkan data dari Tim B dicetak dengan menggunakan objek Map timB.

Pertanyaan

- Dari data tim B, ternyata ada kesalahan pencatatan berat badan yaitu untuk pemain yang memiliki tinggi badan 168, berat badannya adalah 66. Update data

Map untuk tim B.

Source Code

Datasetfutsal.java	
1.	import java.util.*;
2.	
3.	class DataSetFutsal1 {
4.	int[] dataTinggiTimA = { 168, 170, 165, 168, 172,
	170, 169, 165, 171, 166 };
5.	int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,
	66, 56, 72, 56 };
6.	int[] dataTinggiTimB = { 170, 167, 165, 166, 168,
	175, 172, 171, 168, 169 };
7.	int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,
	68, 68, 65, 60 };
8.	
9.	public void BeratBadanSama() {
10.	Set<Integer> BeratBadanSama = new HashSet<>();
11.	int i = 0;
12.	do {
13.	int j = 0;
14.	do {
15.	if (dataBeratBadanTimA[i] ==
	dataBeratBadanTimB[j]) {
16.	BeratBadanSama.add(dataBeratBadanTimA[i]);
17.	}
18.	j++;
19.	} while (j < dataBeratBadanTimB.length);
20.	i++;
21.	} while (i < dataBeratBadanTimA.length);
22.	System.out.println("Pemain yang mempunyai berat
	badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
	"\n"
23.	+ BeratBadanSama);
24.	}
25.	
26.	public void TinggiSama() {
27.	Set<Integer> tinggiSama = new HashSet<>();
28.	int i = 0;
29.	do {
30.	int j = 0;
31.	do {

```

32.         if (dataTinggiTimA[i] ==
dataTinggiTimB[j] &&
!tinggiSama.contains(dataTinggiTimA[i])) {
33.             tinggiSama.add(dataTinggiTimA[i]);
34.         }
35.         j++;
36.     } while (j < dataTinggiTimB.length);
37.     i++;
38. } while (i < dataTinggiTimA.length);
39. System.out.println("Pemain yang mempunyai tinggi
badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
"\n"
40.         + tinggiSama);
41.     }
42.
43.     public void rentangtinggibadan() {
44.         int i = 0;
45.         int mintinggibadan = Integer.MAX_VALUE;
46.         do {
47.             if (dataTinggiTimA[i] < mintinggibadan) {
48.                 mintinggibadan = dataTinggiTimA[i];
49.             }
50.             i++;
51.         } while (i < dataTinggiTimA.length);
52.         i = 0;
53.         do {
54.             if (dataTinggiTimB[i] < mintinggibadan) {
55.                 mintinggibadan = dataTinggiTimB[i];
56.             }
57.             i++;
58.         } while (i < dataTinggiTimB.length);
59.         i = 0;
60.         int maxTinggi = Integer.MIN_VALUE;
61.         do {
62.             if (dataTinggiTimA[i] > maxTinggi) {
63.                 maxTinggi = dataTinggiTimA[i];
64.             }
65.             i++;
66.         } while (i < dataTinggiTimA.length);
67.         i = 0;
68.         do {
69.             if (dataTinggiTimB[i] > maxTinggi) {
70.                 maxTinggi = dataTinggiTimB[i];
71.             }

```



```

72.         i++;
73.     } while (i < dataTinggiTimB.length);
74.     System.out.println("Rentang nilai tinggi badan
kedua tim adalah " + mintinggibadan + " hingga " +
maxTinggi);
75.     }
76.
77.     public void rentangberatbadan() {
78.         int minBeratBadan = Integer.MAX_VALUE;
79.         int i = 0;
80.         do {
81.             if (dataBeratBadanTimA[i] < minBeratBadan) {
82.                 minBeratBadan = dataBeratBadanTimA[i];
83.             }
84.             i++;
85.         } while (i < dataBeratBadanTimA.length);
86.         i = 0;
87.         do {
88.             if (dataBeratBadanTimB[i] < minBeratBadan) {
89.                 minBeratBadan = dataBeratBadanTimB[i];
90.             }
91.             i++;
92.         } while (i < dataBeratBadanTimB.length);
93.         int maxBerat = Integer.MIN_VALUE;
94.         i = 0;
95.         do {
96.             if (dataBeratBadanTimA[i] > maxBerat) {
97.                 maxBerat = dataBeratBadanTimA[i];
98.             }
99.             i++;
100.        } while (i < dataBeratBadanTimA.length);
101.        i = 0;
102.        do {
103.            if (dataBeratBadanTimB[i] > maxBerat) {
104.                maxBerat = dataBeratBadanTimB[i];
105.            }
106.            i++;
107.        } while (i < dataBeratBadanTimB.length);
108.        System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
109.    }
110.
111.    public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()

```

```

{
112.     Set<Integer>    p    =    new HashSet<Integer>();
113.     int i = 0;
114.     do
115.     {
116.         boolean f = false;
117.         int j = 0;
118.         do {
119.             if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
120.                 f = true;
121.                 break;
122.             }
123.             j++;
124.         }
125.         while (j < dataBeratBadanTimA.length);
126.         if (!f) {
127.             p.add(dataBeratBadanTimB[i]);
128.         }
129.         i++;
130.     }
131.     while (i < dataBeratBadanTimB.length);
132.     System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133. }
134.
135. public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
136.     Set<Integer>    p    =    new HashSet<Integer>();
137.     int i = 0;
138.     do {
139.         boolean f = false;
140.         int j = 0;
141.         do {
142.             if      (dataTinggiTimA[i]    == dataTinggiTimB[j])
{
143.                 f = true;
144.                 break;
145.             }
146.             j++;
147.         }
148.         j++;
149.     }
150. }

```

```

151.     while (j < dataTinggiTimB.length);
152.         if (!f) {
153.
154.             p.add(dataTinggiTimA[i]);
155.         }
156.         i++;
157.     }
158.     while (i < dataTinggiTimA.length);
159.     System.out.println("Tinggi badan pada tim
A yang tidak ada pada tim B ada " + p);
160.     }
161.
162.     public void SimpanDatauntukTimAdanTimB() {
163.         Map<Integer, Integer> timA = new HashMap<>();
164.         int i = 0;
165.         do {
166.             timA.put(dataTinggiTimA[i],
dataBeratBadanTimA[i]);
167.             i++;
168.
169.         } while (i < dataTinggiTimA.length);
170.
171.     Map<Integer, Integer> timB = new HashMap<>();
172.     int j = 0;
173.     do
174.     {
175.         timB.put(dataTinggiTimB[j],
dataBeratBadanTimB[j]);
176.         j++;
177.     }
178.     while (j < dataTinggiTimB.length);
179.     System.out.println("Data tinggi dan berat
badan Tim A: " + timA);
180.     System.out.println("Data tinggi berat badan
Tim B: " + timB);
181.
182.     timB.put(168, 66);
183.     System.out.println("Data Tim B setelah
berhasil di update: " + timB);
184.
185.     }
186. }
187. public class datasetfutsal {
188.     public static void main(String[] args) {

```

```

189.         Scanner p = new Scanner(System.in);
190.         DataSetFutsal1 q = new DataSetFutsal1();
191.         // q.TinggiSama();
192.         // q.BeratBadanSama();
193.         // q.rentangtinggibadan();
194.         // q.rentangberatbadan();
195.
196.         //q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
197.         //
198.         q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
199.         q.SimpanDatauntukTimAdanTimB();
200.         p.close();
    }
}

```

Output

The screenshot shows a terminal window with the following output:

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh' & 'C:\Program Files\Java\jdk-11.0.9\bin\java.exe' -cp 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\02422310ae\praktikumkesepuluh_b549c8e3\bin' 'datasetfutsal'
Data tinggi dan berat badan Tim A: {165=56, 166=56, 168=55, 169=66, 170=70, 171=72, 172=60}
Data tinggi berat badan Tim B: {165=59, 166=58, 167=60, 168=65, 169=60, 170=66, 171=68, 172=68, 175=71}
Data Tim B setelah berhasil di update: {165=59, 166=58, 167=60, 168=66, 169=60, 170=66, 171=68, 172=68, 175=71}
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>

```

Below the terminal, a text file named 'Dani Adrian.txt' is open, displaying the following content:

```

Dani Adrian
225150201111009

```

Penjelasan

Jika ingin mengupdate data, maka langsung saja menggunakan syntax `tim_B.put()`, lalu masukkan angka angka baru untuk di update.

Pertanyaan

- Implementasikan Map untuk menyimpan data tinggi badan dan berat badan dari tim A yang tinggi badannya sama dengan tim B.

Source Code

```

datasetfutsal.java
1. import java.util.*;

```

2.	
3.	class DataSetFutsal1 {
4.	int[] dataTinggiTimA = { 168, 170, 165, 168, 172, 170, 169, 165, 171, 166 };
5.	int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70, 66, 56, 72, 56 };
6.	int[] dataTinggiTimB = { 170, 167, 165, 166, 168, 175, 172, 171, 168, 169 };
7.	int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71, 68, 68, 65, 60 };
8.	
9.	public void BeratBadanSama() {
10.	Set<Integer> BeratBadanSama = new HashSet<>();
11.	int i = 0;
12.	do {
13.	int j = 0;
14.	do {
15.	if (dataBeratBadanTimA[i] == dataBeratBadanTimB[j]) {
16.	BeratBadanSama.add(dataBeratBadanTimA[i]);
17.	}
18.	j++;
19.	} while (j < dataBeratBadanTimB.length);
20.	i++;
21.	} while (i < dataBeratBadanTimA.length);
22.	System.out.println("Pemain yang mempunyai berat badan sama ada " + BeratBadanSama.size() + ", yaitu:" + "\n"
23.	+ BeratBadanSama);
24.	}
25.	
26.	public void TinggiSama() {
27.	Set<Integer> tinggiSama = new HashSet<>();
28.	int i = 0;
29.	do {
30.	int j = 0;
31.	do {
32.	if (dataTinggiTimA[i] == dataTinggiTimB[j] && !tinggiSama.contains(dataTinggiTimA[i])) {
33.	tinggiSama.add(dataTinggiTimA[i]);
34.	}
35.	j++;

```

36.         } while (j < dataTinggiTimB.length);
37.         i++;
38.     } while (i < dataTinggiTimA.length);
39.     System.out.println("Pemain yang mempunyai tinggi
    badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
    "\n"
40.         + tinggiSama);
41. }
42.
43. public void rentangtinggibadan() {
44.     int i = 0;
45.     int mintinggibadan = Integer.MAX_VALUE;
46.     do {
47.         if (dataTinggiTimA[i] < mintinggibadan) {
48.             mintinggibadan = dataTinggiTimA[i];
49.         }
50.         i++;
51.     } while (i < dataTinggiTimA.length);
52.     i = 0;
53.     do {
54.         if (dataTinggiTimB[i] < mintinggibadan) {
55.             mintinggibadan = dataTinggiTimB[i];
56.         }
57.         i++;
58.     } while (i < dataTinggiTimB.length);
59.     i = 0;
60.     int maxTinggi = Integer.MIN_VALUE;
61.     do {
62.         if (dataTinggiTimA[i] > maxTinggi) {
63.             maxTinggi = dataTinggiTimA[i];
64.         }
65.         i++;
66.     } while (i < dataTinggiTimA.length);
67.     i = 0;
68.     do {
69.         if (dataTinggiTimB[i] > maxTinggi) {
70.             maxTinggi = dataTinggiTimB[i];
71.         }
72.         i++;
73.     } while (i < dataTinggiTimB.length);
74.     System.out.println("Rentang nilai tinggi badan
    kedua tim adalah " + mintinggibadan + " hingga " +
    maxTinggi);
75. }

```

```

76.
77.     public void rentangberatbadan() {
78.         int minBeratBadan = Integer.MAX_VALUE;
79.         int i = 0;
80.         do {
81.             if (dataBeratBadanTimA[i] < minBeratBadan) {
82.                 minBeratBadan = dataBeratBadanTimA[i];
83.             }
84.             i++;
85.         } while (i < dataBeratBadanTimA.length);
86.         i = 0;
87.         do {
88.             if (dataBeratBadanTimB[i] < minBeratBadan) {
89.                 minBeratBadan = dataBeratBadanTimB[i];
90.             }
91.             i++;
92.         } while (i < dataBeratBadanTimB.length);
93.         int maxBerat = Integer.MIN_VALUE;
94.         i = 0;
95.         do {
96.             if (dataBeratBadanTimA[i] > maxBerat) {
97.                 maxBerat = dataBeratBadanTimA[i];
98.             }
99.             i++;
100.        } while (i < dataBeratBadanTimA.length);
101.        i = 0;
102.        do {
103.            if (dataBeratBadanTimB[i] > maxBerat) {
104.                maxBerat = dataBeratBadanTimB[i];
105.            }
106.            i++;
107.        } while (i < dataBeratBadanTimB.length);
108.        System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
109.    }
110.
111.    public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
    {
112.        Set<Integer>    p    =    new HashSet<Integer>();
113.        int i = 0;
114.        do
115.        {
116.            boolean f = false;

```

```

117.         int j = 0;
118.         do {
119.             if (dataBeratBadanTimB[i] ==
dataBeratBadanTimA[j]) {
120.                 f = true;
121.                 break;
122.             }
123.             j++;
124.         }
125.         while (j < dataBeratBadanTimA.length);
126.         if (!f) {
127.             p.add(dataBeratBadanTimB[i]);
128.         }
129.         i++;
130.     }
131.     while (i < dataBeratBadanTimB.length);
132.     System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133. }
134.
135. public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
136.     Set<Integer> p = new HashSet<Integer>();
137.     int i = 0;
138.     do {
139.         boolean f = false;
140.         int j = 0;
141.         do {
142.             if (dataTinggiTimA[i] == dataTinggiTimB[j])
{
143.                 f = true;
144.                 break;
145.             }
146.             j++;
147.         }
148.         while (j < dataTinggiTimB.length);
149.         if (!f) {
150.             p.add(dataTinggiTimA[i]);
151.         }
152.         i++;
153.     }
154.     while (i < dataTinggiTimB.length);
155.     System.out.println("Tinggi badan pada tim B yang
tidak ada pada tim A ada " + p);
156. }

```



```

157.         }
158.         while (i < dataTinggiTimA.length);
159.         System.out.println("Tinggi badan pada tim
A yang tidak ada pada tim B ada " + p);
160.     }
161.
162.     public void SimpanDatauntukTimAdanTimB() {
163.         Map<Integer, Integer> timA = new HashMap<>();
164.         int i = 0;
165.         do {
166.             timA.put(dataTinggiTimA[i],
dataBeratBadanTimA[i]);
167.             i++;
168.
169.         } while (i < dataTinggiTimA.length);
170.
171.         Map<Integer, Integer> timB = new HashMap<>();
172.         int j = 0;
173.         do
174.         {
175.             timB.put(dataTinggiTimB[j],
dataBeratBadanTimB[j]);
176.             j++;
177.         }
178.         while (j < dataTinggiTimB.length);
179.         System.out.println("Data tinggi dan berat
badan Tim A: " + timA);
180.         System.out.println("Data tinggi berat badan
Tim B: " + timB);
181.
182.         timB.put(168, 66);
183.         System.out.println("Data Tim B setelah
berhasil di update: " + timB);
184.
185.         Map<Integer, Integer> x = new
HashMap<>();
186.         Iterator<Map.Entry<Integer, Integer>>
iteratorA = timA.entrySet().iterator();
187.         do {
188.
189.         Map.Entry<Integer, Integer> entryA = iteratorA.next();
190.         int tinggi = entryA.getKey();
191.
192.         if (timB.containsKey(tinggi)) {

```

```

193.         x.put(tinggi, entryA.getValue());
194.     }
195. }
196.     while (iteratorA.hasNext());
197.         System.out.println("Data tinggi dan berat badan
dari tim A yang sama dengan tim B adalah " + x);
198.     }
199. }
200.
201. public class datasetfutsal {
202.     public static void main(String[] args) {
203.         Scanner p = new Scanner(System.in);
204.         DataSetFutsal1 q = new DataSetFutsal1();
205.         // q.TinggiSama();
206.         // q.BeratBadanSama();
207.         // q.rentangtinggibadan();
208.         // q.rentangberatbadan();
209.
210.         //q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
211.         //
212.         q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
213.         q.SimpanDatauntukTimAdanTimB();
214.         p.close();
215.     }
216. }

```

Output

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> c;; cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh'; & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\1\c58c2d2cb\redhat.java\jdt_ws\praktikumkesepuluh_b549c8e3\bin' 'datasetfutsal'
Data tinggi dan berat badan Tim A: {165=56, 166=56, 168=55, 169=66, 170=70, 171=72, 172=60}
Data tinggi berat badan Tim B: {165=59, 166=58, 167=60, 168=65, 169=60, 170=66, 171=68, 172=68, 175=71}
Data Tim B setelah berhasil di update: {165=59, 166=58, 167=60, 168=66, 169=60, 170=66, 171=68, 172=68, 175=71}
Data tinggi dan berat badan dari tim A yang sama dengan tim B adalah {165=56, 166=56, 168=55, 169=66, 170=70, 171=72, 172=60}
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>

```

Dani Adrian.txt

Dani Adrian
225150201111009

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Penjelasan

Method `simpanData TimA TimB()`:

- Mencari data tinggi dan berat badan yang sama antara tim A dan tim B.
- Data yang sama kemudian disimpan dalam objek x, yang merupakan objek dari kelas `HashMap` dengan tipe data `Integer` sebagai key dan value.
- Method ini menggunakan iterator untuk mengakses setiap elemen dalam objek `timA` (yang tidak didefinisikan dalam kode yang diberikan).
- Jika nilai tinggi dalam `timA` juga ada dalam `timB`, maka pasangan key-value ditambahkan ke objek x.

Class `datasetfutsal`

- Objek `Data_SetFutsal` dengan nama q dibuat dan method `simpanData_TimA_TimB()` dari objek tersebut dipanggil.

Pertanyaan

- d. Dari data tim A yang mempunyai tinggi badan sama dengan tim B tadi, pelatih memutuskan untuk hanya mengambil pemain yang memiliki tinggi badan 168 keatas, sehingga pemain dengan tinggi badan kurang dari 168 dinyatakan di eliminasi dari tim. Update data Map tersebut sesuai dengan kondisi terakhir.

Source Code

datasetfutsal.java	
1.	<code>import java.util.*;</code>
2.	
3.	<code>class DataSetFutsal1 {</code>
4.	<code> int[] dataTinggiTimA = { 168, 170, 165, 168, 172,</code>
	<code>170, 169, 165, 171, 166 };</code>
5.	<code> int[] dataBeratBadanTimA = { 50, 60, 56, 55, 60, 70,</code>
	<code>66, 56, 72, 56 };</code>
6.	<code> int[] dataTinggiTimB = { 170, 167, 165, 166, 168,</code>
	<code>175, 172, 171, 168, 169 };</code>
7.	<code> int[] dataBeratBadanTimB = { 66, 60, 59, 58, 58, 71,</code>
	<code>68, 68, 65, 60 };</code>
8.	
9.	<code> public void BeratBadanSama() {</code>

```

10.         Set<Integer> BeratBadanSama = new HashSet<>();
11.         int i = 0;
12.         do {
13.             int j = 0;
14.             do {
15.                 if (dataBeratBadanTimA[i] ==
dataBeratBadanTimB[j]) {
16.                     BeratBadanSama.add(dataBeratBadanTimA[i]);
17.                 }
18.                 j++;
19.             } while (j < dataBeratBadanTimB.length);
20.             i++;
21.         } while (i < dataBeratBadanTimA.length);
22.         System.out.println("Pemain yang mempunyai berat
badan sama ada " + BeratBadanSama.size() + ", yaitu:" +
"\n"
23.             + BeratBadanSama);
24.     }
25.
26.     public void TinggiSama() {
27.         Set<Integer> tinggiSama = new HashSet<>();
28.         int i = 0;
29.         do {
30.             int j = 0;
31.             do {
32.                 if (dataTinggiTimA[i] ==
dataTinggiTimB[j] &&
!tinggiSama.contains(dataTinggiTimA[i])) {
33.                     tinggiSama.add(dataTinggiTimA[i]);
34.                 }
35.                 j++;
36.             } while (j < dataTinggiTimB.length);
37.             i++;
38.         } while (i < dataTinggiTimA.length);
39.         System.out.println("Pemain yang mempunyai tinggi
badan yang sama ada " + tinggiSama.size() + ", yaitu:" +
"\n"
40.             + tinggiSama);
41.     }
42.
43.     public void rentangtinggibadan() {
44.         int i = 0;
45.         int mintinggibadan = Integer.MAX_VALUE;

```

```

46.         do {
47.             if (dataTinggiTimA[i] < mintinggibadan) {
48.                 mintinggibadan = dataTinggiTimA[i];
49.             }
50.             i++;
51.         } while (i < dataTinggiTimA.length);
52.         i = 0;
53.         do {
54.             if (dataTinggiTimB[i] < mintinggibadan) {
55.                 mintinggibadan = dataTinggiTimB[i];
56.             }
57.             i++;
58.         } while (i < dataTinggiTimB.length);
59.         i = 0;
60.         int maxTinggi = Integer.MIN_VALUE;
61.         do {
62.             if (dataTinggiTimA[i] > maxTinggi) {
63.                 maxTinggi = dataTinggiTimA[i];
64.             }
65.             i++;
66.         } while (i < dataTinggiTimA.length);
67.         i = 0;
68.         do {
69.             if (dataTinggiTimB[i] > maxTinggi) {
70.                 maxTinggi = dataTinggiTimB[i];
71.             }
72.             i++;
73.         } while (i < dataTinggiTimB.length);
74.         System.out.println("Rentang nilai tinggi badan
kedua tim adalah " + mintinggibadan + " hingga " +
maxTinggi);
75.     }
76.
77.     public void rentangberatbadan() {
78.         int minBeratBadan = Integer.MAX_VALUE;
79.         int i = 0;
80.         do {
81.             if (dataBeratBadanTimA[i] < minBeratBadan) {
82.                 minBeratBadan = dataBeratBadanTimA[i];
83.             }
84.             i++;
85.         } while (i < dataBeratBadanTimA.length);
86.         i = 0;
87.         do {

```

```

88.         if (dataBeratBadanTimB[i] < minBeratBadan) {
89.             minBeratBadan = dataBeratBadanTimB[i];
90.         }
91.         i++;
92.     } while (i < dataBeratBadanTimB.length);
93.     int maxBerat = Integer.MIN_VALUE;
94.     i = 0;
95.     do {
96.         if (dataBeratBadanTimA[i] > maxBerat) {
97.             maxBerat = dataBeratBadanTimA[i];
98.         }
99.         i++;
100.    } while (i < dataBeratBadanTimA.length);
101.    i = 0;
102.    do {
103.        if (dataBeratBadanTimB[i] > maxBerat) {
104.            maxBerat = dataBeratBadanTimB[i];
105.        }
106.        i++;
107.    } while (i < dataBeratBadanTimB.length);
108.    System.out.println("Rentang nilai berat badan
kedua tim adalah " + minBeratBadan + " hingga " +
maxBerat);
109.    }
110.
111. public void BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA()
    {
112.     Set<Integer>    p    =    new HashSet<Integer>();
113.     int i = 0;
114.     do
115.     {
116.         boolean f = false;
117.         int j = 0;
118.         do {
119.             if      (dataBeratBadanTimB[i]    ==
dataBeratBadanTimA[j]) {
120.                 f = true;
121.                 break;
122.             }
123.             j++;
124.         }
125.         while (j < dataBeratBadanTimA.length);
126.         if (!f) {
127.             p.add(dataBeratBadanTimB[i]);

```

```

128. }
129.     i++;
130. }
131.     while (i < dataBeratBadanTimB.length);
132.         System.out.println("Berat badan pada tim B yang
tidak ada pada tim A ada " + p);
133.     }
134.
135. public void
TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB() {
136.     Set<Integer>    p    =    new HashSet<Integer>();
137.     int i = 0;
138.     do {
139.         boolean f = false;
140.         int j = 0;
141.         do {
142.             if      (dataTinggiTimA[i]  == dataTinggiTimB[j])
{
143.                 f = true;
144.                 break;
145.             }
146.             j++;
147.         }
148.         while (j < dataTinggiTimB.length);
149.         if (!f) {
150.             p.add(dataTinggiTimA[i]);
151.             i++;
152.         }
153.         while (i < dataTinggiTimA.length);
154.         System.out.println("Tinggi badan pada tim
A yang tidak ada pada tim B ada " + p);
155.     }
156.
157.     public void SimpanDatauntukTimAdanTimB() {
158.         Map<Integer, Integer> timA = new HashMap<>();
159.         int i = 0;
160.         do {
161.             timA.put(dataTinggiTimA[i],
dataBeratBadanTimA[i]);

```

```

167.         i++;
168.
169.         } while (i < dataTinggiTimA.length);
170.
171. Map<Integer, Integer> timB = new HashMap<>();
172.     int j = 0;
173.     do
174.     {
175.         timB.put(dataTinggiTimB[j],
dataBeratBadanTimB[j]);
176.         j++;
177.     }
178.         while (j < dataTinggiTimB.length);
179.         System.out.println("Data tinggi dan berat
badan Tim A: " + timA);
180.         System.out.println("Data tinggi berat badan
Tim B: " + timB);
181.
182.         timB.put(168, 66);
183.         System.out.println("Data Tim B setelah
berhasil di update: " + timB);
184.
185.         Map<Integer, Integer> x      =      new
HashMap<>();
186.         Iterator<Map.Entry<Integer, Integer>>
iteratorA = timA.entrySet().iterator();
187.         do {
188.
189. Map.Entry<Integer, Integer> entryA = iteratorA.next();
190.         int tinggi = entryA.getKey();
191.
192.         if (timB.containsKey(tinggi)) {
193.             x.put(tinggi, entryA.getValue());
194.         }
195.     }
196.         while (iteratorA.hasNext());
197.         System.out.println("Data tinggi dan berat badan
dari tim A yang sama dengan tim B adalah " + x);
198. Map<Integer, Integer> y      =      new HashMap<>();
199.         Iterator<Map.Entry<Integer, Integer>> iteratorB =
timA.entrySet().iterator();
200.         do {
201.             Map.Entry<Integer, Integer> entryB =
iteratorB.next();

```



```

202.         int tinggi = entryB.getKey();
203.         if (tinggi >= 168) {
204.             y.put(tinggi, entryB.getValue());
205.         }
206.     } while (iteratorB.hasNext());
207.     System.out.println("Data Tim A setelah eliminasi
tinggi badan kurang dari 168 adalah " + y);
208. }
209. }
210.
211. public class datasetfutsal {
212.     public static void main(String[] args) {
213.         Scanner p = new Scanner(System.in);
214.         DataSetFutsal1 q = new DataSetFutsal1();
215.         // q.TinggiSama();
216.         // q.BeratBadanSama();
217.         // q.rentangtinggibadan();
218.         // q.rentangberatbadan();
219.
220.         //q.TinggiBadanTIMA_tidakAdaPada_TinggiBadanTimB();
221.         //
222.         q.BeratBadanTIMB_tidakAdaPada_BeratBadanTIMA();
223.         q.SimpanDatauntukTimAdanTimB();
224.         p.close();
225.     }
226. }

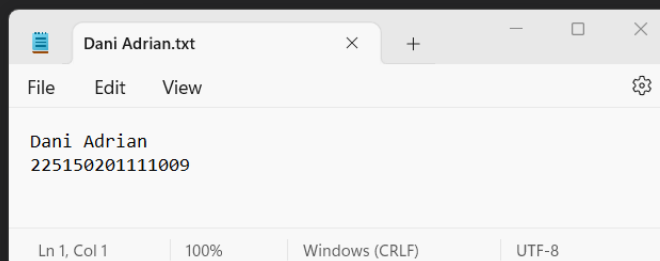
```

Output

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh'; & 'C:\Program F
a.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage
ca58c2d2cb\redhat.java\jdt_ws\praktikumkesepuluh_b549c8e3\bin' 'datasetfutsal'
Data tinggi dan berat badan Tim A: {165=56, 166=56, 168=55, 169=66, 170=70, 171=72, 172=60}
Data tinggi berat badan Tim B: {165=59, 166=58, 167=60, 168=65, 169=60, 170=66, 171=68, 172=68, 175=71}
Data Tim B setelah berhasil di update: {165=59, 166=58, 167=60, 168=66, 169=60, 170=66, 171=68, 172=68, 175=71}
Data tinggi dan berat badan dari tim A yang sama dengan tim B adalah {165=56, 166=56, 168=55, 169=66, 170=70, 171=72, 172=60}
Data Tim A setelah eliminasi tinggi badan kurang dari 168 adalah {168=55, 169=66, 170=70, 171=72, 172=60}
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkesepuluh>

```



Penjelasan

Method simpanData TimA TimB

Pada Method ini, terdapat variabel `y` yang merupakan objek dari kelas `HashMap` yang digunakan untuk menyimpan data tinggi dan berat badan tim A setelah dilakukan eliminasi tinggi badan yang kurang dari 168.

Dilakukan iterasi menggunakan iterator `iteratorB` untuk mengakses setiap entry (pasangan kunci dan nilai) dalam objek `tim_A`.

Setiap entry dicek untuk tinggi badan yang lebih besar atau sama dengan 168.

Jika memenuhi kondisi, pasangan kunci dan nilai tersebut ditambahkan ke dalam objek `y`.

Loop akan terus berlanjut selama masih terdapat entry selanjutnya dalam iterator. Setelah loop selesai, hasil data tinggi tim A yang telah dieliminasi ditampilkan ke layar.

Class dataSetFutsal

Objek `Data_SetFutsal` dengan nama `q` dibuat dan Method `simpanData_TimA_TimB` dipanggil.