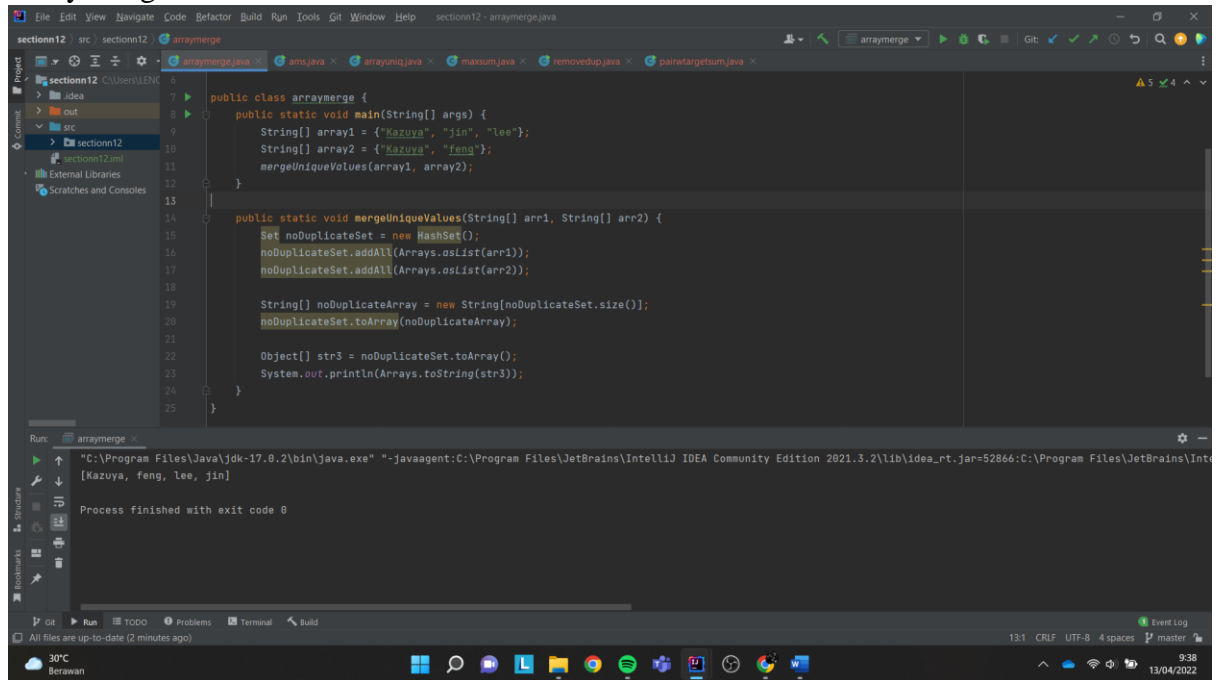


Tugas section 12 & 13

Nurul Muthia Anggraeni Yasin

Kelas E QE

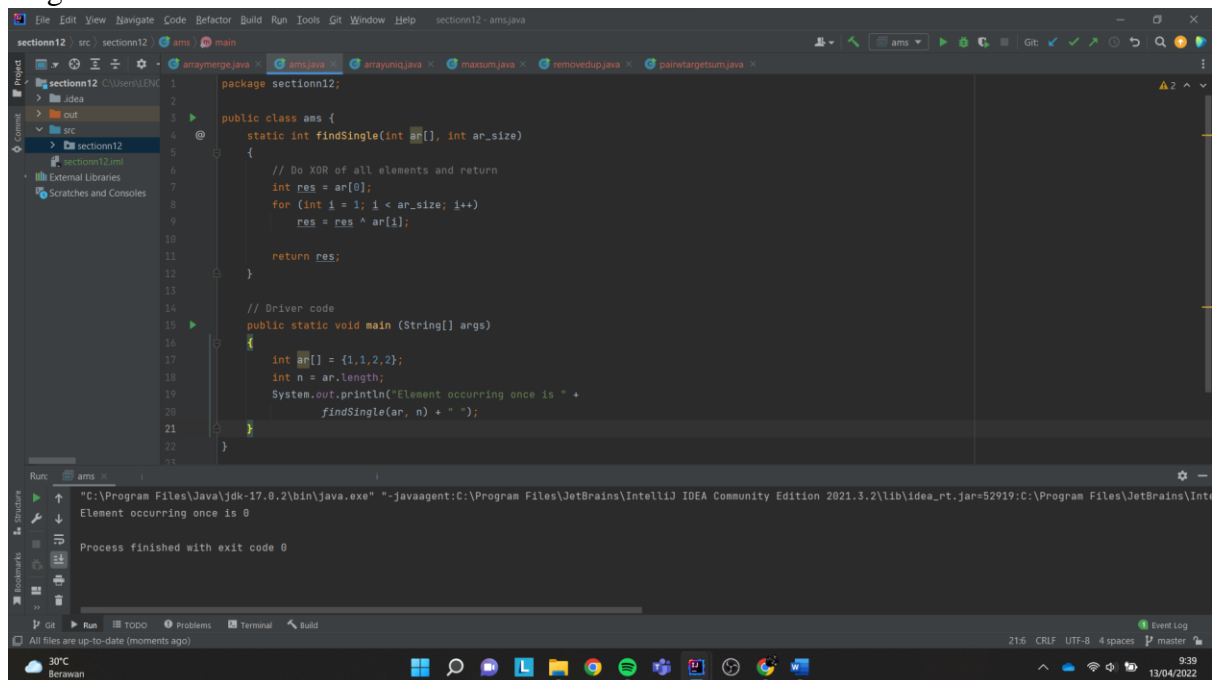
- Array merge



```
public class arraymerge {  
    public static void main(String[] args) {  
        String[] array1 = {"Kazuya", "jin", "lee"};  
        String[] array2 = {"Kazuya", "feng"};  
        mergeUniqueValues(array1, array2);  
    }  
  
    public static void mergeUniqueValues(String[] arr1, String[] arr2) {  
        Set noDuplicateSet = new HashSet();  
        noDuplicateSet.addAll(Arrays.asList(arr1));  
        noDuplicateSet.addAll(Arrays.asList(arr2));  
  
        String[] noDuplicateArray = new String[noDuplicateSet.size()];  
        noDuplicateSet.toArray(noDuplicateArray);  
  
        Object[] str3 = noDuplicateSet.toArray();  
        System.out.println(Arrays.toString(str3));  
    }  
}
```

Run: arraymerge  
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea\_rt.jar=52866:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin --add-opens=java.base/java.util=ALL-UNNAMED  
[Kazuya, feng, lee, jin]  
Process finished with exit code 0

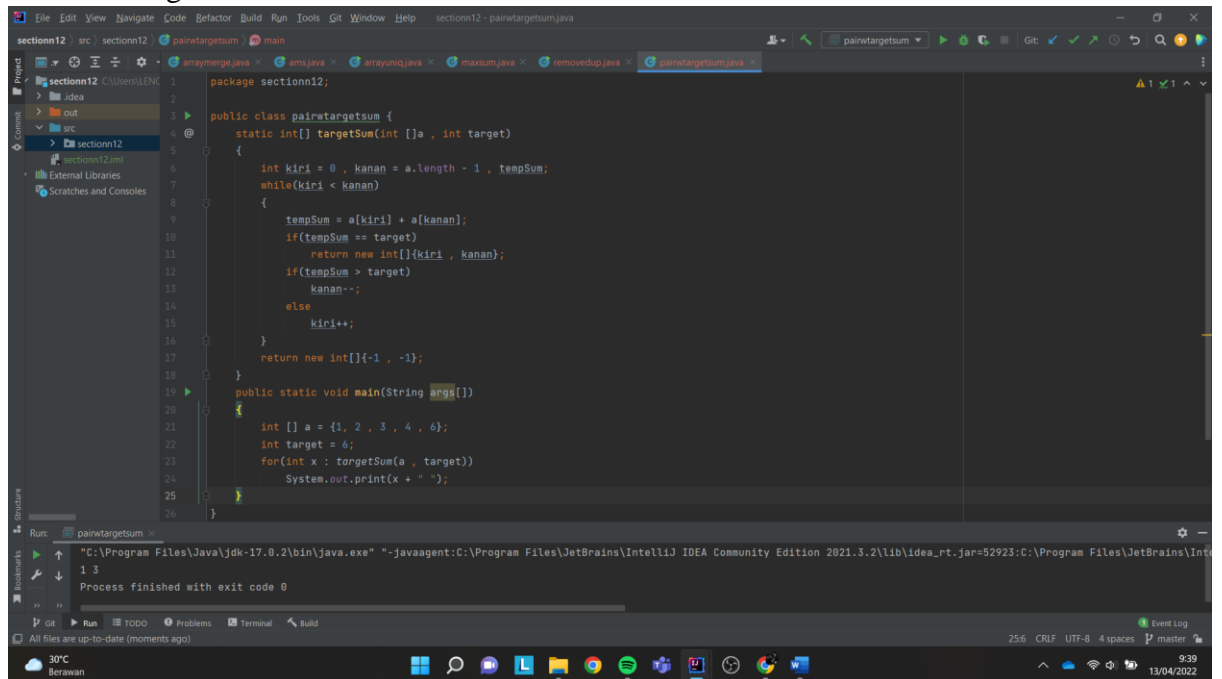
- Angka muncul sekali



```
package section12;  
  
public class ams {  
    static int findSingle(int ar[], int ar_size)  
    {  
        // Do XOR of all elements and return  
        int res = ar[0];  
        for (int i = 1; i < ar_size; i++)  
            res = res ^ ar[i];  
  
        return res;  
    }  
  
    // Driver code  
    public static void main (String[] args)  
    {  
        int ar[] = {1,1,2,2};  
        int n = ar.length;  
        System.out.println("Element occurring once is " +  
            findSingle(ar, n) + " ");  
    }  
}
```

Run: ams  
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea\_rt.jar=52919:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin --add-opens=java.base/java.util=ALL-UNNAMED  
Element occurring once is 0  
Process finished with exit code 0

- Pair with target sum



The screenshot shows the IntelliJ IDEA IDE with a Java project named 'section12'. The file 'pairWithTargetSum.java' is open, showing the following code:

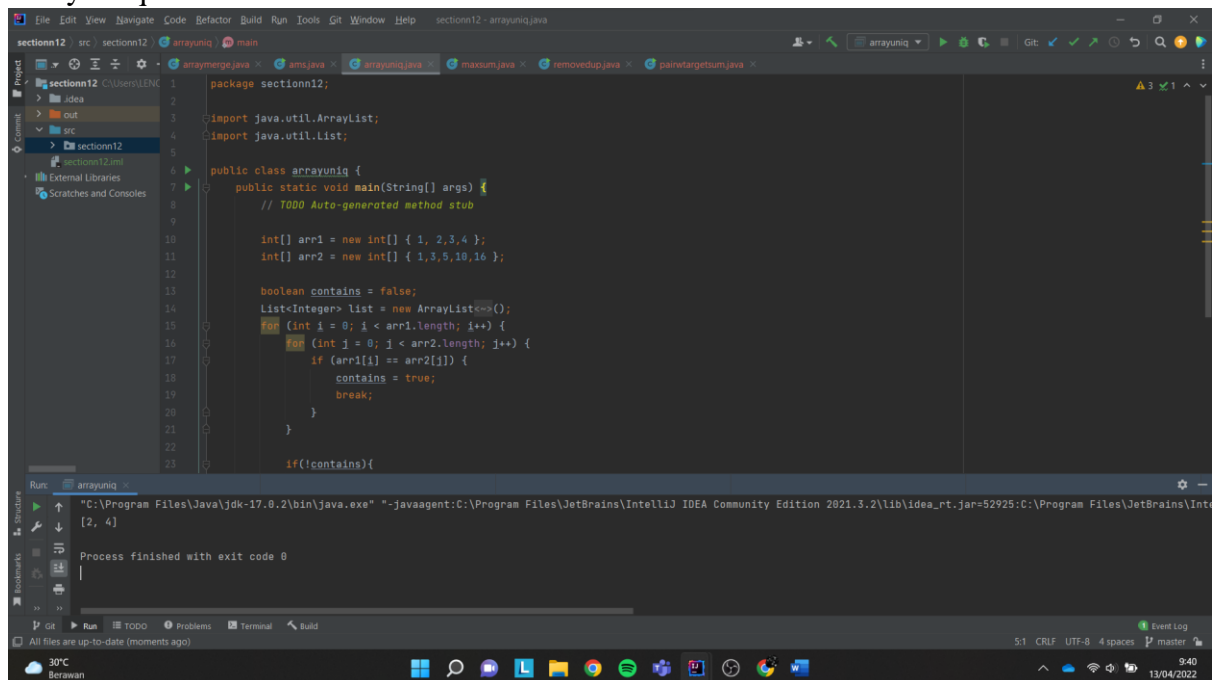
```
package section12;

public class pairWithTargetSum {
    static int[] targetSum(int []a , int target)
    {
        int kiri = 0 , kanan = a.length - 1 , tempSum;
        while(kiri < kanan)
        {
            tempSum = a[kiri] + a[kanan];
            if(tempSum == target)
                return new int[]{kiri , kanan};
            if(tempSum > target)
                kanan--;
            else
                kiri++;
        }
        return new int[]{-1 , -1};
    }

    public static void main(String args[])
    {
        int [] a = {1 , 2 , 3 , 4 , 6};
        int target = 6;
        for(int x : targetSum(a , target))
            System.out.print(x + " ");
    }
}
```

The Run window at the bottom shows the output: '1 3'. The process finished with exit code 0.

- Array unique



The screenshot shows the IntelliJ IDEA IDE with a Java project named 'section12'. The file 'arrayUnique.java' is open, showing the following code:

```
package section12;

import java.util.ArrayList;
import java.util.List;

public class arrayUnique {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

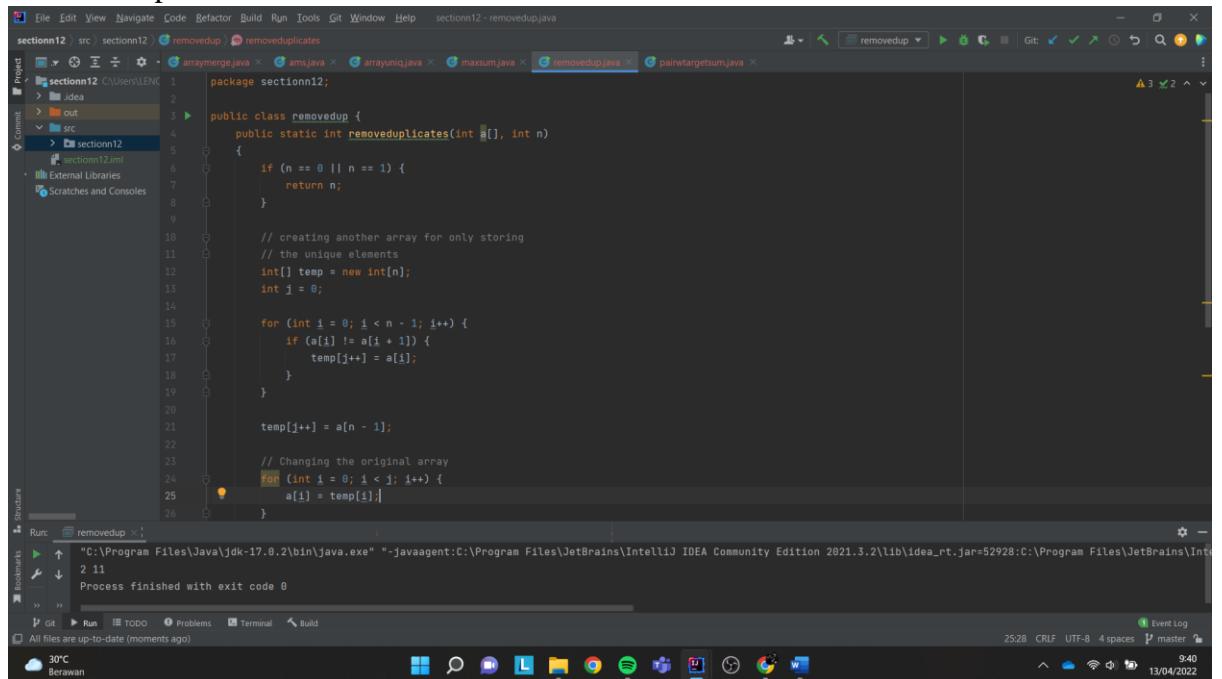
        int[] arr1 = new int[] { 1, 2, 3, 4 };
        int[] arr2 = new int[] { 1, 3, 5, 10, 16 };

        boolean contains = false;
        List<Integer> list = new ArrayList<>();
        for (int i = 0; i < arr1.length; i++) {
            for (int j = 0; j < arr2.length; j++) {
                if (arr1[i] == arr2[j]) {
                    contains = true;
                    break;
                }
            }
        }

        if(!contains){
            list.add(arr1[i]);
        }
    }
}
```

The Run window at the bottom shows the output: '[2, 4]'. The process finished with exit code 0.

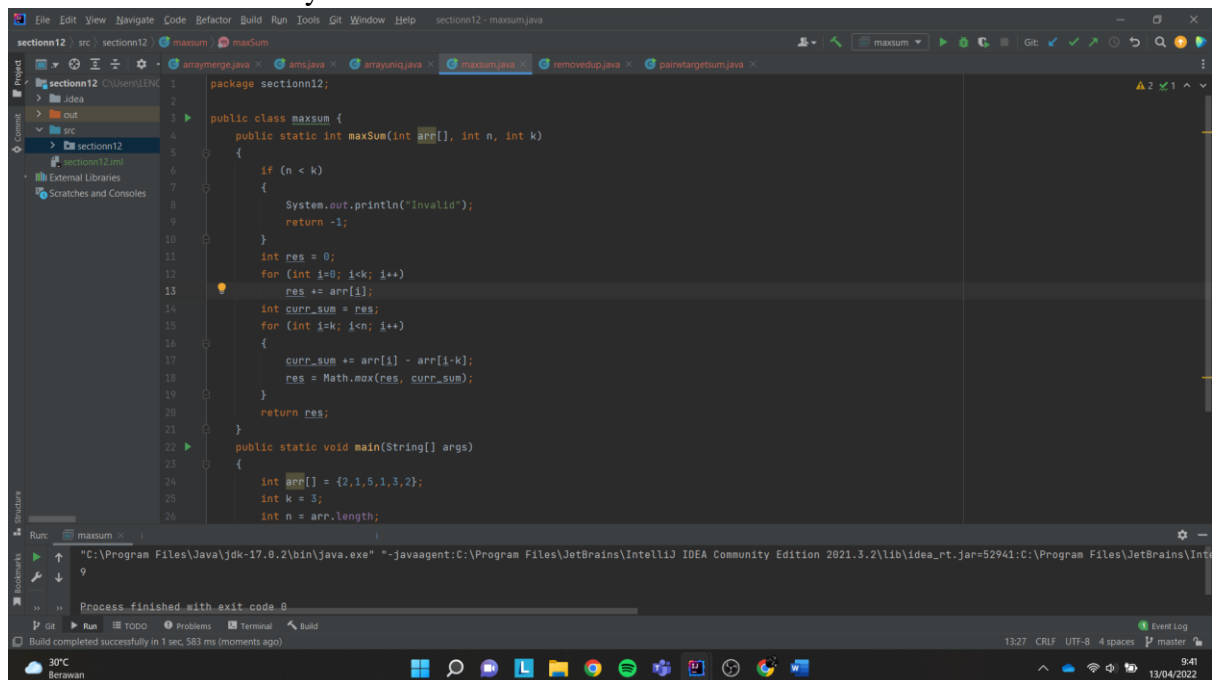
- Remove duplicates



```
1 package section12;
2
3 public class removedup {
4     public static int removeduplicates(int a[], int n)
5     {
6         if (n == 0 || n == 1) {
7             return n;
8         }
9
10        // creating another array for only storing
11        // the unique elements
12        int[] temp = new int[n];
13        int j = 0;
14
15        for (int i = 0; i < n - 1; i++) {
16            if (a[i] != a[i + 1]) {
17                temp[j++] = a[i];
18            }
19        }
20
21        temp[j++] = a[n - 1];
22
23        // Changing the original array
24        for (int i = 0; i < j; i++) {
25            a[i] = temp[i];
26        }
27    }
28 }
```

Run: removedup x; 2 11 Process finished with exit code 0

- Maximum sum subarray of size K



```
1 package section12;
2
3 public class maxsum {
4     public static int maxSum(int arr[], int n, int k)
5     {
6         if (n < k)
7         {
8             System.out.println("Invalid");
9             return -1;
10        }
11
12        int res = 0;
13        for (int i = 0; i < k; i++)
14            res += arr[i];
15
16        int curr_sum = res;
17        for (int i = k; i < n; i++)
18        {
19            curr_sum += arr[i] - arr[i - k];
20            res = Math.max(res, curr_sum);
21        }
22        return res;
23    }
24
25    public static void main(String[] args)
26    {
27        int arr[] = {2, 1, 5, 1, 3, 2};
28        int k = 3;
29        int n = arr.length;
30    }
31 }
```

Run: maxsum x; 9 Process finished with exit code 0